2018

Hardware Security and VLSI Design Optimization

Hao Xue
Wright State University

Follow this and additional works at: https://corescholar.libraries.wright.edu/etd_all

Part of the Engineering Commons

Repository Citation
https://corescholar.libraries.wright.edu/etd_all/2206

This Dissertation is brought to you for free and open access by the Theses and Dissertations at CORE Scholar. It has been accepted for inclusion in Browse all Theses and Dissertations by an authorized administrator of CORE Scholar. For more information, please contact library-corescholar@wright.edu.
HARDWARE SECURITY AND VLSI DESIGN OPTIMIZATION

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

By

HAO XUE
M.S.Eg., Wright State University, USA, 2013
B.S., Taiyuan University of Technology, China, 2010

2018

Wright State University

___________________________
Saiyu Ren, Ph.D.
Dissertation Director

___________________________
Arnab K. Shaw, Ph.D.
Director, Electrical Engineering Ph.D. Program

___________________________
Barry Milligan, Ph.D.
Interim Dean of the Graduate School

Committee on Final Examination

___________________________
Saiyu Ren, Ph.D.

___________________________
Ray Siferd, Ph.D.

___________________________
Marty Emmert, Ph.D.

___________________________
Marian Kazimierczuk, Ph.D.

___________________________
Yan Zhuang, Ph.D.
Abstract


Microelectronic circuit is ubiquitous component of modern electrical devices. The increasing complexity and professionalism of phases in microelectronic supply chain bring more global cooperation to integrated circuit (IC) production. Therefore, providing a secure environment for microelectronic circuit design does not ensure the integrity of the hardware since any participator of IC fabrication has the opportunity to implant a malicious alteration in original IC design. Especially overseas chip-fabrication is a vital potential threat for national defense products. In theory, anyone who has access to fabrication process can tamper with the original design, with the potential to change function, modify parametric properties or even have confidential information transmitted to the attacker. The surreptitious modification of an IC is denoted as Hardware Trojan (HT). To address the issue of providing robust and reliable IC products, this dissertation proposes HT detection techniques which are based on HT activation and side-channel analysis. Simulation results show that the proposed technique can detect HT with areas that are 0.013% of the host-circuitry. Combinational use of multiple detection techniques will facilitate detection probability. Also, low power-delay-product (PDP) VLSI design is considered for optimizing parametric overhead of detection circuit. Simulation results indicate that the proposed VLSI design optimization techniques can improve PDP of dynamic and static CMOS circuits by up to 61.9% and 49.9%, respectively.
# TABLE OF CONTENTS

## 1 INTRODUCTION

1.1 Background ................................................................. 1

1.2 Hardware Security.......................................................... 2

  1.2.1 Hardware Trojan Structure .......................................... 2

  1.2.2 Characterization of Hardware Trojans ............................ 3

    1.2.2.1 Activation Characteristics ................................... 4

    1.2.2.2 Physical Characteristics .................................... 4

    1.2.2.3 Action Characteristics ...................................... 6

  1.2.3 Hardware Trojan Countermeasures.................................. 7

    1.2.3.1 Pre-Silicon Hardware Trojan Countermeasures .............. 8

    1.2.3.2 Post-Silicon Hardware Trojan Countermeasures ............. 10

    1.2.3.3 In-Field Hardware Trojan Countermeasures ................ 13

1.3 VLSI Design Optimization .............................................. 13

1.4 Dissertation Objective.................................................. 14

1.5 Dissertation Organization ............................................. 14

## 2 Hardware Trojan Detection Efficiency Improvement .............. 15

2.1 Introduction ....................................................................... 15

2.2 Probability-Increase-Circuit (PIC) Insertion.......................... 16

  2.2.1 PIC for Rare Signal when $P_0 \gg P_1$ ............................... 16

    2.2.1.1 PIC Design .................................................. 16

    2.2.1.2 PIC Overhead ................................................ 19

  2.2.2 PIC for Rare Signal when $P_0 \ll P_1$ ............................... 21

    2.2.2.1 PIC Design .................................................. 21

    2.2.2.2 PIC Overhead ................................................ 23

2.3 PIC Insertion Algorithm .................................................. 24

2.4 Experimental Result .................................................... 25

2.5 Conclusion ................................................................. 31

## 3 Timing Analysis-based Hardware Trojan Detection .................. 32

3.1 Introduction ..................................................................... 32
LIST OF FIGURES

Fig. 1.1 HT Structure ........................................................................................................... 2
Fig. 1.2 HT Classification .................................................................................................... 3
Fig. 1.3 Inverter Gate without (a) and with dopant HT (b) .............................................. 6
Fig. 1.4 IC Process ................................................................................................................ 8
Fig. 1.5 HT Countermeasures .............................................................................................. 8
Fig. 1.6 FSM of Five States Implemented by Two Sub-FSM .......................................... 10
Fig. 2.1 8-Bit AND Gate, Signal Probability Label \( P_0, P_1 \) ......................................... 17
Fig. 2.2 Probability Increase Circuit for \( P_0 \gg P_1 \) ......................................................... 18
Fig. 2.3 8-Bit AND Gate with PIC, Signal Probability Label \( P_0, P_1 \) ....................... 19
Fig. 2.4 8-Bit OR Gate, Probability Label \( P_0, P_1 \) ...................................................... 21
Fig. 2.5 Probability Increase Circuit for \( P_0 \ll P_1 \) ......................................................... 22
Fig. 2.6 8-Bit OR Gate with PIC, Probability Label \( P_0, P_1 \) ....................................... 23
Fig. 2.7 PIC Insertion Flow ............................................................................................... 25
Fig. 2.8 64-Bit Binary Comparator .................................................................................... 26
Fig. 3.1 Block Diagram of HT Detection Circuit Using PUF ........................................ 34
Fig. 3.2 Signals in Trojan-Free (TF) and Trojan-Attacked (TA) Testing-Path ............ 36
Fig. 3.3 Timing Analysis-based HT Detection Algorithm .............................................. 37
Fig. 3.4 HT Detection Block Diagram ............................................................................ 38
Fig. 3.5 Testing-Path in CUT ......................................................................................... 39
Fig. 3.6 Operation Waveform of HT Detection in Corner \( TT \) (typical-typical), (a) Trojan-Free (TF), (b) Trojan-Attacked (TA) ...................................................... 40
Fig. 3.7 Feedback-AND-OR (FAO) Gate ........................................................................ 42
Fig. 3.8 HT Location in CUT, (a) In-Path, (b) By-Path, (c) Off-Path ......................... 43
Fig. 3.9 Ideal Signals in Trojan-Free (TF) and Trojan-Attacked (TA) Testing Path ....... 45
Fig. 3.10 CUT with fan-out \( I \) (a), and fan-out \( n \) (b) ...................................................... 46
Fig. 3.11 Ring-Oscillator Scheme ..................................................................................... 48
Fig. 3.12 Detection of RCAs with In-Path HT, (a) 4-Bit, (b) 8-Bit, (c) 16-Bit RCA ............50
Fig. 3.13 Detectable In-Path HT/Testing-Path Size .................................................................54
Fig. 3.14 Detection of RCAs with By-Path HT, (a) 4-Bit, (b) 8-Bit, (c) 16-Bit RCA .............56
Fig. 3.15 Detectable By-Path HT/Testing-Path Size .................................................................58
Fig. 3.16 Digital FIR Filter of Order n .......................................................................................60
Fig. 3.17 FPGA Board (XC6VLX240T) ...................................................................................61
Fig. 3.18 FPGA ChipScope Organization .................................................................................61
Fig. 3.19 FPGA Implementation Result of FIR Filter ...............................................................62
Fig. 3.20 Low-Power Carry-Select-Adder .................................................................................64
Fig. 4.1 Partitioned CUT ............................................................................................................70
Fig. 4.2 Static Power Deviation of Gate GI .............................................................................76
Fig. 4.3 Overall Flow of Self-Reference-based HT Detection .................................................77
Fig. 4.4 Sequential Circuit Structure .......................................................................................79
Fig. 4.5 Partitioned CUT1 .........................................................................................................80
Fig. 4.6 lclk Deviation in CUT1 ..............................................................................................81
Fig. 4.7 ISCAS-89 Benchmarks (a) S27, (b) S344, and ISCAS-85 Benchmarks (c) c499, (d) c6288, (e) c5315 ........................................................................................................83
Fig. 4.8 Threshold of Clock-Tree Scaling Factor Deviation in Benchmarks .......................84
Fig. 4.9 Partitioned CUT2 .........................................................................................................92
Fig. 4.10 Simulation Result of CUT2 with Different Size of Gate Group .............................94
Fig. 5.1 Block Diagram of CUT ...............................................................................................102
Fig. 5.2 Segment 1 of CUT with HT (a) In-path, (b) By-path, (c) Off-path .........................102
Fig. 5.3 Power Gating ............................................................................................................106
Fig. 5.4 Structure of Power Gated HT ....................................................................................107
Fig. 6.1 Conventional Dynamic CMOS Circuit, (a) Non-Inverted, (b) Inverted, (c) Operation Modes ..............................................................................................................110
Fig. 6.2 Dynamic 2-Input NAND Gate, (a) Conventional, (b) Up-Footed .........................112
Fig. 6.3 Dynamic 2-Input AND Gate (a) without (b) with Node-Discharger

Fig. 6.4 Operation Waveform of Dynamic 2-Input AND Gate, (a) Ideal (b) Simulation
LIST OF TABLES

Table 2.1 Activation Time of Rare Signal in CUTs ........................................................................................................19
Table 2.2 Overhead of PIC for Rare Signal when $P_0 \gg P_1$ .........................................................................................20
Table 2.3 Activation Time of Rare Signal in CUTs ........................................................................................................23
Table 2.4 Overhead of PIC for Rare Signal when $P_0 \ll P_1$ ..........................................................................................24
Table 2.5 Performance of CUT and PIC ..........................................................................................................................30
Table 3.1 Paths Operation in FAO Gate ..........................................................................................................................42
Table 3.2 HT Detection Parameters with Variable Calibrated Delay Time .................................................................45
Table 3.3 Delay of CUT with different fan-out .................................................................................................................46
Table 3.4 HT detection for CUT with different number of fan-outs ............................................................................47
Table 3.5 In-Path HT Detection in RCAs .........................................................................................................................51
Table 3.6 Detectable In-Path HT/Testing-Path Size at 90% Detection Probability ............................................................53
Table 3.7 By-Path HT Detection in RCAs .........................................................................................................................57
Table 3.8 Detectable By-Path HT/Testing-Path Size at 90% Detection Probability ............................................................58
Table 3.9 Power Consumption of CUT & HT Detection Circuit ......................................................................................60
Table 3.10 Detectable HT Size in FPGA Boards .............................................................................................................63
Table 3.11 HT Detection Result on Benchmarks ...........................................................................................................65
Table 3.12 Comparison of HT Detection Methods .........................................................................................................66
Table 4.1 Static Power of Gates in CUT1 ........................................................................................................................87
Table 4.2 Variation Scaling Factors in Trojan-Free CUT1 ..............................................................................................90
Table 4.3 Variation Scaling Factors of Clock-Tree ...........................................................................................................90
Table 4.4 Variation Scaling Factors in CUT2 with Different Gate Group Size ............................................................93
Table 4.5 HT Detection Result on Benchmarks .............................................................................................................96
Table 4.6 Comparison of Different Trojan Detection Method .........................................................................................97
Table 5.1 Side-channel Effect of HT in Various Locations ............................................................................................101
Table 5.2 Detectable HT Size Over Host-circuit Size at 90% Detection Probability ....................................................103
Table 5.3 Simulation Results of Timing & Power Analysis-based HT Detection ..........................................................104
Table 5.4 Simulation Results of Power Analysis-based HT Detection with/without HT Activation

Table 6.1 Performance Improvement in Non-Inverted Dynamic Benchmarks

Table 6.2 Performance Improvement in Inverted Dynamic Benchmarks
1 INTRODUCTION

1.1 Background

Microelectronic circuit is ubiquitous component of modern electrical devices. The increasing complexity and professionality of phases in microelectronic supply chain bring more global cooperation to integrated circuit (IC) production. Also outsourcing of IC manufacture to low-cost location is effective to counteract market pressure. Hardware integrity is becoming a serious topic with more and more entities involving in IC production, since any participator of IC fabrication has the opportunity to implant a hardware Trojan (malicious alteration of genuine IC) to original IC design. The hardware Trojan insertion has been considered as a serious threat in the past decade [1]-[6]. About 3,000 years ago, after 10 years fruitless conquering the city of Troy, the Greek army pretended to sail away leaving behind a huge wooden horse that actually with some Greek solider inside. The Trojans pulled the horse into Troy. Greek forces slipped out of the horse and opened the gate for the sailing back Greek army, then ended the war [7]. Hardware Trojan (HT) is similar that is a tiny alteration of original circuit design that may distorter prescribed IC function.

A HT can remain active or stay dormant until activation criteria happens. The HT activation signal may be an external signal (e.g. interact with off-chip signals by antenna, sensor) or an internal signal (e.g. logic, operation timing). An active HT may change the functionality when compared to the original design [8], tamper with parametric properties [9], or even have confidential information transmitted to the attacker [10].

Detection of such HT is difficult due to the following reasons:

1) Reverse engineering and physical inspection of post-fab chips are time and money costly for IC supply chain, also it is not security guarantee for the
remaining ICs.

2) Compare to the tremendously increasing complexity of modern IC, the embedded HT is typically with relatively very tiny size that is hard to be revealed.

3) Due to stealthy nature, normally HT is triggered by a rare signal and remain dormant mode in most lifetime. It will reduce the operating effect of HT on host-chip thereby even further enhancing the difficulty to distinguish it.

4) With feature size decreasing, process and environmental variations have emerged more impact on circuit parameters. HT detections using simple parameter-analysis have become more and more ineffective.

5) Unknown location and form of HT blinds its detection specification.

1.2 Hardware Security

HT structure, characterization, and countermeasures are introduced in this section.

1.2.1 Hardware Trojan Structure

A typical HT structure consists of trigger logic and payload, as shown in Fig. 1.1 [11]. The trigger logic monitors HT trigger signal which is a set of signals activating HT circuit thereby with payload effected. To maintain stealthy nature of HT in testing mode, the fundamental in this case is that trigger signal has low controllability while HT effect has low observability, that means trigger signal should be relatively very rare case compare to other signals on chip [12].

![Hardware Trojan Structure](image)

Fig. 1.1 HT Structure
1.2.2 Characterization of Hardware Trojans

An unknown HT may be triggered by various signal, consist of various form of HT circuit, and act as various payload. That is a serious puzzle for chip designers to counteract the potential HT insertion. To properly evaluate HTs and propose effective countermeasures, HT is classified and analyzed in this section. Typically, HT is characterized by its activation mechanism (referred as trigger), physical design (referred as HT circuit), and action effect (referred as payload), shown in Fig. 1.2 [5]. Note that HT may hybrid of these classification, for example, activated by multiple characteristics.

![HT Classification Diagram](image-url)

Fig. 1.2 HT Classification
1.2.2.1 Activation Characteristics

HT activation characteristics refer to the criteria that trigger HT thereafter proceeding its effect. Activation characteristics are classified into *internally activated* and *externally activated*.

An internally activated TH can be subdivided into *always-on* and *conditions* based. Always-on represents the HT being active at all times, for example, a reduced wire in clock-tree. Clock-tree is used to distribute clock signal to clocked elements at the same time. A chip that is modified in this way will suffer from clock skew that may producing errors or fails every time the wire is used intensely. Adversary may deploy always-on HT in area with rare operation to remain silent. Condition based activation monitors one or more signals inside IC as trigger. It can be either analog signal (e.g., temperature, delay, timing), or digital signal (e.g., Boolean logic function).

The opposite is externally activated TH. There can be malicious logic inside a chip, that uses an antenna or sensors receiving HT trigger signal from outside of IC. Due to the isolation of IC die and the outer signal, this trigger must be analog signal (e.g., radio). In this case, the adversary may even communicate with the inserted HT in real-time.

1.2.2.2 Physical Characteristics

HT physical characteristics refer to the physical design of HT. The physical HT characteristics include *distribution*, *structure*, *type*, and *size*.

HT distribution represents the layout density of HT on chip, that consists of *tight* and *loose*. Tight distribution HT has its components relatively close in chip layout, whereas loose distribution HT distributes its components loosely across the chip layout. The reason why HT may distribute loose or tight is: 1) it is dependent on free layout area in original chip. The adversary is unwilling to redesign the existing layout in host-
chip, because change of original design may affect its operation parameters (e.g., modified layout wire may change signal delivery time) that is prone to be an abnormal testing. 2) variable location of HT trigger signal and HT effect area on chip to meet HT design specification. Note that HT with loose distribution will scatter its parameter effect on host-chip thereby benefit its silent characteristic. However, communication of loose distributed components requires increased wires that attendant increased timing and power effect on host-chip.

HT structure is subclassified into layout-change and layout-same, indicating host-chip’s original layout is changed and unchanged, respectively. Though change of original layout may increase the abnormal detection probability, HT designers sometimes still force to do so, because: 1) HT is eager to squeeze into a specific area utilizing the operations in it (e.g., use the signal in this area as trigger or modify the function in this area). 2) scattering HT in free layout areas requires long wires for communication of each partition of HT. The increased wires further change the delay and power characteristics of host-chip, breaking HT’s silence. Therefore, changing the original layout to harbor HT with tight distribution may benefit some HT designs. Also hiding technology (e.g., power gating) can be used to facilitate stealthy nature of HT.

HT type describes the implementation type of HT that is partitioned to functional type and parametric type. Functional type represents the HT causing addition or deletion of transistors or gates, which is relatively popular due to its various possible functionalities, whereas parametric type indicates the HT modifying original circuit, like thinning of wires, weakening of flip-flops or transistors, subjecting the chip to radiation, or using focused ion beams (FIB) to reduce the reliability of a chip. A parametric type HT example is presented as in Fig. 1.3 [13]. It is an inverter attacked by dopant HT reversing dopant polarity in specific area to change the behavior of gate.
In original inverter, shown in Fig. 1.3 (a), the PMOS (top) transistor consists of n-well, positively doped source and drain (p-dopant), while the NMOS transistor (bottom) includes p-well, negatively doped source and drain (n-dopant). With dopant HT, as shown in Fig. 1.3 (b), the p-dopant in PMOS is replaced by n-dopant, so power source \( V_{DD} \) remains connecting to source, n-well, then drain; in the source of NMOS, p-dopant instead of n-dopant is applied thereby ground \( V_{ss} \) is isolated from drain regardless of its gate input. As a result, the dopant HT attacked inverter will permanently output \( V_{DD} \).

Fig. 1.3 Inverter Gate without (a) and with dopant HT (b)

HT size accounts for the number of component that have been added, deleted or altered. HT with large size looks like is easier to be detected, however HT with small size may lose ability monitoring complicated rare trigger that increases its activation probability, eventually harming silence of HT.

### 1.2.2.3 Action Characteristics

HT action characteristics refer to the effect of HT once its triggered. The HT action characteristics include transit-information, modify-specification, and modify-function.

A HT with transit-information action will transit confidential information from host-chip to adversary. For example, in 2014 January, the magazine of New York Times
reported that the Quantum program has helped intelligence agencies to receive data from computers miles away by using tiny circuits inserted surreptitiously into system transiting a covert channel of radio wave [14].

Modify-specification refer to HT attacking the parameter properties (e.g., delay) of host-chip. It may change or disable the expected host-chip operation. For example, HT can change designed chip operation by exhausting scarce resources such as bandwidth, computation, and battery power. A HT can be designed to consume excess battery energy by preventing circuits from going to sleep; alter stored data by over-writing an existing value with a random value; or even disable operation by simply isolate partial or all power supply from circuits. In 2007, a Syrian radar failed to warn an air strike, later that is reported be responsible by a backdoor built in the system chip [15].

Modify-function refer to HT modify designed function of host-chip by adding extra logic components or bypassing original circuits. Because HT with modify-specification action only has ability to change existing components, it can lead to operation failure. However, HT with modify-function action can build its own logic thereby implement more possible unexpected operations.

1.2.3 Hardware Trojan Countermeasures

The goal of HT countermeasure is to ensure that the IC product used by customer is authentic. IC production process includes design, fabrication, testing, and operation, shown in Fig. 1.4. The best way to prevent HT-attack on IC is to tightly control the process from beginning to end. Typically, IC design mode and testing mode are conducted by IC designers, thereafter IC is deployed in operation mode with customer. IC manufacture fabrication is considered as the only untrusted mode (out control of IC designer) that is normally outsourced to dedicated IC foundries. Many HT
countermeasures are published in the past decade [1][3]. They can be classified into three types: pre-silicon, post-silicon, and in-field HT countermeasures, shown in Fig. 1.5, which are refer to IC design, testing, and operation mode, respectively.

![Fig. 1.4 IC Process](image)

![Fig. 1.5 HT Countermeasures](image)

### 1.2.3.1 Pre-Silicon Hardware Trojan Countermeasures

Preventing HT insertion is considered in pre-silicon testing mode before fabrication. To insert HT, an attacker needs two requirements: 1) understand original IC design, or at least familiar with partial IC design that will be attacked; 2) have enough resource on-chip to serve HT, like trigger signal, layout area, power source. Preventing HT insertion consists of camouflage and exhausting resource, that counteracting the two HT insertion requirements, respectively.

1. **Camouflage**

Camouflage, as the name indicates, is a strategy to hide the genuine components
and IC logic, thereby obfuscating HT design when interface to host-IC. Camouflaging components design benefits inhibiting IC attackers from extracting genuine gate-level netlist. It can be done by altering layout of standard cell [16][17], adding dummy contacts, or useless connections between layers [18][19]. To hide IC logic, original circuit is accompanied with a locking circuit which is transparent when a specific key is applied. The increasing complexity of distinguishing locking circuit with unknown key impedes the design of practical HT. In [20], authors added an XOR gate and an inverter to standard half-adder that will work functionally only when prescribed key-bit meets criteria. Note that the modification of original logic or standard components may degrade the IC operation performance.

2. **Exhausting Resource**

To avoid detection, normally designers squeeze HT size to minimize its effect on host-chip. Furthermore, sophisticated HT design may use exiting circuit on chip for HT’s trigger signal generator or even activity. Exhausting free resources on chip will occupy the elements that may be used by HT. Some examples are presented below.

**Layout Space:** Since design tools are typically conservative in circuit layout placement, there are always free layout spaces in IC design. Filler cells are used to fill up these spaces. Attackers may replace the filler cells with HT without affecting original function cells. To counteract this case, we can modify all unused silicon to be looks like functional circuit which is denoted as functional filler [21]. Genuine circuitry accompanied with realistic functional filler will embarrass HT designers when finding either available HT inserting spot or practical HT trigger. However, the functional fillers will increase power consumption thereby generating heat in limited spaces further degrading IC performance.

**Power Source:** A HT consume power for both monitoring trigger in dormant
mode and effecting payload in active mode. Constraining power supply to barely satisfy normal operational requirement is effective in squeezing power use for HT. It can be done through partitioning IC into segments which use separate power rails. Each partitioned power rail is designed with capability to deliver sufficient current for the corresponding circuit segment. Thereafter, inserted HT may rob normal used power supply, resulting error function.

State: Occupy or eliminate unused states of sequential circuit. An example is introduced with a finite-state machine (FSM) which is a mathematical model of computation that can be in one of finite number of states. An FSM of five states, requires three \((2^3 > 5)\) state elements with binary encoding [22]. The three \((2^3 − 5 = 3)\) unused states may be used as a sequential HT trigger signal without effect on host circuit. In this case, two sub-FSM with smaller states can generate five states, shown in Fig. 1.6, meanwhile eliminate useless states that may be used by HT designer. Sub-FSM \(FSM_1\) consists of four states \((S0, S1, S2, S3)\), while \(FSM_2\) consists of two states \((S'0, S'1)\). They have one overlapping state \((S3/S'0)\).

Fig. 1.6 FSM of Five States Implemented by Two Sub-FSM

As aforementioned, both camouflage and exhausting resource may degrade IC performance. Balance of IC authentication and operation performance is a critical concern for designers.

1.2.3.2 Post-Silicon Hardware Trojan Countermeasures

HT detection of a chip is conducted in post-silicon testing mode before deployment. Detecting HT is a serious challenge due to its stealthy nature, inordinately
large number of possible instances and large variety in structure and operation mode. Traditional fault testing techniques are insufficient in this extraordinary situation. A manufacturing fault happens at a random position whereas malicious changes are well placed to avoid detection. The HT detection is subclassified into destructive and non-destructive detections.

Destructive HT detection refers to reverse engineering which is an invasive and destructive form of analyzing an IC chip. Reverse engineers grind away IC die layer by layer and microphoto the netlists. Sometimes it is even possible to attach a probe to measure voltages while the IC is still operational. This technique has ability to reveal complete hardware of the IC chip. Thereafter its used to detect HT by comparing the netlists of testing IC with original design [23]. Destructive HT detection is possible to achieve 100% detection probability, whereas attendant extreme high timing and money-cost. Also, this method, as the name indicates, can only be used on a sample group of ICs with no guarantee that untested ICs are Trojan-free. However destructive HT detection can be used to characterize the 100% genuine IC (e.g., power/timing fingerprint) to facilitate other detection methods.

Non-destructive HT detection consists of HT activation, functional testing, and side-channel analysis.

1. **HT Activation**

Due to stealthy nature of HT, typically it uses rare cases as trigger to avoid accidental activation or detection during testing mode. Staying in dormant can avoid unnecessary function change and tremendously reduce the differential operating parameters caused by HT, eventually limit the HT detection probability. Triggering HT will therefore benefit the efficiency of traditional detection techniques, like HT detection based on functional testing or side-channel analysis. In [24], authors proposed
a dummy scan flip-flop inserting to nodes with low transition probability (may use as trigger signal) for reducing transition time. Thereafter all nodes on host-chip could have transition probability more than a specific level, and rare signals will appear in greater opportunity.

2. Functional Testing

HT detection based on functional testing compares functional response of testing IC with the designed ones. HT presence is declared when it affects functional response during testing mode. In [25], authors detect HT using a randomization-based technique to probabilistically compare the operational function of testing IC with the designed function. Note that a dormant HT will minimize its effect, especially functional response, on host-chip. HT activation technology is effective in constrain time to run functional testing-based HT detection.

3. Side-Channel Analysis

Side-channel analysis is popular in HT detection, because HT insertion affects the host-circuit’s side-channel parameters consist of timing, power consumption, and electromagnetic emanation (EM). HT can be revealed by distinguishing the differential side-channel parameters of the attacked circuit. Also, HT activation technology may wake HT, if exist, increasing its activity to facilitate side-channel analysis.

Embedded HT adds extra capacitance, resulting in more charging and discharging delays to HT affected paths. Authors in [26] chose a number of clean (free of HTs) ICs and ran a variety of input patterns to record specific delays of paths to establish key delay fingerprints. The testing ICs are simulated under the same input patterns and verified by comparing timing parameters with genuine fingerprint.

Embedded HT adds current paths and loads to the original circuit, that result in extra power consumption on wires and gates in HT affected area. In [27], authors
distinguish the HT-attacked IC by comparing the power traces of testing ICs with genuine IC. Statistical methods are used to improve the detection probability.

Operation of electrical circuits is attendant emitting signals like magnetic and electric fields. Those signals can be analyzed to conclude the operational state and data in the testing circuit. Authors in [28] propose a detection technique for hard-to-detect points by using an electromagnetic probe to trace electromagnetic emanation for each transition step point in testing chip.

1.2.3.3 In-Field Hardware Trojan Countermeasures

The aforementioned classes of HT detection are applied prior to deployment, however a sophisticated HT with parameter-muted design (e.g., power gating,) may have ability to bypass all pre-deployment detections thereafter releasing malicious effect in-field. A self-detection design is necessary in field to compensate this loophole, such that HT countermeasure can take place when dormant HT is activated in customer use. A practical self-detection circuit should have three specifications: 1) low-overhead, 2) work friendly with host-chip at speed, 3) react with HT to stop hurting chip-user (warning, turn off power, etc.). In [29], a real-time trust evaluation framework is proposed to monitor IC global power consumption in field and warn abnormal operations if happens. In [30], a concurrent error detection (CED) technique is used in field to detect malicious signal derived from HT.

1.3 VLSI Design Optimization

Circuit operation at high frequencies while consuming low power is one of the most important characteristics in designing ICs. Also, parametric overhead of host-chip associate with HT countermeasures is a serious concern for IC designers. The added components will bring extra current path and load to original design. VLSI design optimization is thereby studied to minimize HT countermeasure overhead.
1.4 Dissertation Objective

There are two objectives in this dissertation:

- Develop, implement and assess effective HT detection techniques with high detection probability, low overhead, and tunable detection probability to accommodate different security requirements of circuit designs.
- Develop, implement, and assess the performance of VLSI design optimization techniques that can be used to benefit high performance CMOS circuit design, especially constraining HT detection circuit overhead.

1.5 Dissertation Organization

The rest of this dissertation is organized as follows. Chapter 2 introduces a design increasing probability of rare signals which may be the trigger of HT to facilitate the HT detection efficiency. Chapter 3 and 4 present HT detection methodologies based on timing and power-analysis, respectively. Combinational use of HT detection methods and its benefits are introduced in chapter 5. VLSI design optimization for dynamic and static CMOS circuit design are proposed in chapter 6 and 7, respectively. Eventually, chapter 8 presents conclusion and future work.
2 Hardware Trojan Detection Efficiency Improvement

This chapter introduces a probability-increase-circuit (PIC) increasing probability of rare signal which may be the trigger of hardware Trojan (HT) to facilitate the HT detection efficiency. Preliminary results for a standard HT example show a reduction in time for HT activation of over 95% with modest increase in power, size, and delay overhead. The discussion in this chapter is substantially drawn from [31], where we first reported the development and evaluation of this technique.

2.1 Introduction

To have a reliable microelectronic product, HT detection has become a critical concern for researches in the past decade.

An embedded HT will add extra timing and power consumption to the HT affected area. As long as the differences of timing and power consumption of circuit-under-test (CUT) and genuine chip are not coverable by process and environmental variations, the CUT is declared as HT-attacked. Efficient input patterns may magnify HT impact on operating parameters of host-chip that facilitates distinguishing HT from host-chip. Reference [32] applies specific input patterns to lower host circuit activity while keep HT active to maximize the HT contribution in power consumption. Resistance and capacitance impact of HT on the timing of host-chip are magnified using specific input patterns in [26]. Current integration is proposed to detect HT and analyze its location in [33]. Also input patterns are used to increase the switching probability of host chip for improving the efficiency of detection.

As seen, in prior works, researchers exploit the difference of operating parameters on genuine ICs and testing ICs as a basis for HT detection methodologies. However due to the stealthy nature of HT, typically its designed to bypass or disable the security fence of a system: it stays in dormant mode in most lifetime that tremendously reduces
the differential operating parameters and the HT trigger signal is associated with rare input patterns. Therefore, triggering HT will benefit the efficiency of traditional detection techniques. In [24], Salmani proposed a dummy scan flip-flop inserting to nodes with low transition probability for reducing transition time. Thereafter all nodes on host-chip could have transition probability more than a specific level, and rare signals will appear in greater opportunity. However, the proposed dummy scan flip-flop is with multiple times extra power consumption and 1-2 orders of magnitude larger area-overhead that is unacceptable sacrifice in microelectronic circuit industry. In this work, we introduce a much simpler design achieving the same goal, but only with negligible power and area-overhead. A circuit named probability-increase-circuit (PIC) is embedded in rare signal nodes to increase the probability of the rare signal. Geometric distribution is applied to model average number of primary input patterns that required to convert signal on nodes [34].

2.2 Probability-Increase-Circuit (PIC) Insertion

Signal transition of nodes on chip is induced by change of previous level output, while transition of first level components is derived from change of primary inputs. Due to the unpredictable location of HT, the transition probability of all nodes on chip should be raised to a certain level to avoid rare signal trigger, thereby reducing HT active time. For nodes on chip, we assume $P_0$ and $P_1$ are probability of having signal ‘0’ and ‘1’, respectively. Idle node contains two situations: $P_0 \gg P_1$ and $P_0 \ll P_1$ that are introduced in this section.

2.2.1 PIC for Rare Signal when $P_0 \gg P_1$

2.2.1.1 PIC Design

A standard 8-bit AND gate is presented in Fig. 2.1, which output ‘I’ is a rare signal example that may be used as HT trigger. The trigger signal (output of AND gate)
is activated (high voltage) while all inputs of AND gate \((IN_1-IN_8)\) are ‘1’. Signal probability of nodes in Fig. 2.1 is labeled as format \(P_0, P_1\) with assumption that primary inputs have equal probability of values. As shown in Fig. 2.1, \textit{HT trigger signal} is passive in \(255/256\) rate, and active in \(1/256\) rate, that is \(P_0 \gg P_1\). Based on geometric distribution, the average number of primary input patterns that required for triggering output of 8-bit AND gate is

\[ N = \frac{1}{P_a + P_p} - 1 = \frac{1}{\frac{255}{256} \cdot \frac{1}{256}} - 1 = 256 \quad (2.1) \]

where \(P_p\) and \(P_a\) are probability of 8-bit AND gate output being passive (low voltage) and active (high voltage), respectively. As seen, average 256 input patterns are required to trigger the rare signal (output of 8-bit AND gate).

![Diagram of 8-bit AND Gate](image)

Fig. 2.1 8-Bit AND Gate, Signal Probability Label \((P_0, P_1)\)

A PIC is developed as shown in Fig. 2.2 to increase the probability of signal transition when \(P_0 \gg P_1\). In Fig. 2.2, \textit{NODE} is the rare signal node input; \textit{OUT} is the node output; \textit{TS} (testing switch) is switch of signal transition increase, node signal transition probability is increased when \(TS\) is ‘I’; input \textit{SW} (square wave) alternately provides ‘I’ and ‘0’ in a certain frequency. To turn on \textit{PIC}, input \textit{TS} is switched to ‘I’,
CMOS switch is on, then square wave propagates from $SW$ to OR gate. The rare signal node is connected to another input ($NODE$) of OR gate. Thereafter $OUT$ receives the same value as $NODE$ when $SW$ is ‘0’, while $OUT$ remains ‘1’ when SW is ‘1’. Because $SW$ is square wave and $OUT$ is always ‘1’ when $SW$ is ‘1’, $OUT$ should be ‘1’ in halftime. In another halftime, $SW$ switches to ‘0’, $OUT$ is virtually remaining ‘0’ when $P_0 \gg P_1$.

Eventually, $P_1$ and $P_0$ of signal $OUT$ are much closer compared to the original design. Based on simulation in CMOS IBM 90nm technology, power consumption of PIC in passive and active mode are 32.71nW and 440.3nW, respectively.

![Diagram](Fig. 2.2 Probability Increase Circuit for $P_0 \gg P_1$)

The 8-bit AND gate with PIC is present in Fig. 2.3, as seen which, the probability of values in $OUT$ is much closer ($735/1024$, $289/1024$). The average number of primary input patterns that required for triggering signal $OUT$ is

$$N = \frac{1}{P_p} - 1 = \frac{1}{\frac{735}{1024}} - 1 = 4$$

(2.2)

As seen, only 4 rather than 256 input patterns are required to trigger the rare signal $OUT$ (output of 8-bit AND gate).
2.2.1.2 PIC Overhead

As introduced in chapter 0, Trojan attack is normally happened in untrusted fabrication, and its concerned to be detected in post-fab testing mode. Therefore, the proposed PIC will only be turned on in testing mode while remains off in field. For power-overhead calculation in customer use, the power consumption of PIC in dormant mode is taken into account while the host-circuit is active. 8-bit, 16-bit and 32-bit AND gates are used as CUT simulating with/without PIC. The activation time of rare signal in CUTs is shown in Table 2.1, where CUTs operate at input frequency of 1GHz. As seen, the rare signal activation time is tremendous reduced by PIC for all CUTs with varied sizes.

Table 2.1 Activation Time of Rare Signal in CUTs

<table>
<thead>
<tr>
<th>CUT</th>
<th>CUT without PIC</th>
<th>CUT with PIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-bit AND</td>
<td>256ns</td>
<td>4ns</td>
</tr>
<tr>
<td>16-bit AND</td>
<td>66us</td>
<td>4ns</td>
</tr>
<tr>
<td>32-bit AND</td>
<td>4.3s</td>
<td>4ns</td>
</tr>
</tbody>
</table>
Overhead of PIC in CUTs is shown in Table 2.2, where circuit size is estimated by number of transistors; power is simulated in Cadence with CMOS IBM 90nm technology when PIC is off (dormant mode); delay is circuit critical path worst delay. As seen, even the parameter overhead of PIC is huge for relatively small CUT, it is decreased linearly with the increase of corresponding CUT size. In other words, the PIC overhead can be restricted to a practical level by limiting the number of PICs in a chip. It may be implemented by: 1) decrease the targeting transition probability level; 2) apply less PIC in security-robust portion of circuit (e.g., critical path) compared to security-sensitive portion of circuit (e.g., memory). Critical path is security-robust, because any extra connection on critical path will slow chip operating frequency, which can be easily revealed in post-fab functional test. Memory circuit is security-sensitive, because the key data stored in memory is prone to be a target for attackers.

Table 2.2 Overhead of PIC for Rare Signal when $P_0 \gg P_1$

<table>
<thead>
<tr>
<th></th>
<th>CUT</th>
<th>CUT without PIC</th>
<th>CUT with PIC</th>
<th>Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Size</strong></td>
<td>8-bit AND</td>
<td>42</td>
<td>62</td>
<td>47.6%</td>
</tr>
<tr>
<td></td>
<td>16-bit AND</td>
<td>90</td>
<td>106</td>
<td>17.8%</td>
</tr>
<tr>
<td></td>
<td>32-bit AND</td>
<td>186</td>
<td>202</td>
<td>8.6%</td>
</tr>
<tr>
<td><strong>Power (µW)</strong></td>
<td>8-bit AND</td>
<td>1.568</td>
<td>1.813</td>
<td>15.6%</td>
</tr>
<tr>
<td></td>
<td>16-bit AND</td>
<td>3.155</td>
<td>3.194</td>
<td>1.2%</td>
</tr>
<tr>
<td></td>
<td>32-bit AND</td>
<td>6.328</td>
<td>6.368</td>
<td>0.6%</td>
</tr>
<tr>
<td><strong>Delay (ps)</strong></td>
<td>8-bit AND</td>
<td>94</td>
<td>132</td>
<td>40.4%</td>
</tr>
<tr>
<td></td>
<td>16-bit AND</td>
<td>129</td>
<td>167</td>
<td>29.5%</td>
</tr>
<tr>
<td></td>
<td>32-bit AND</td>
<td>176</td>
<td>214</td>
<td>21.6%</td>
</tr>
</tbody>
</table>
2.2.2 PIC for Rare Signal when $P_0 \ll P_1$

2.2.2.1 PIC Design

A standard 8-bit OR gate is presented in Fig. 2.4, which output ‘0’ is a rare signal example that may be used as HT trigger. The trigger signal (output of OR gate) is activated (low voltage) while all inputs of OR gate ($IN_1$-$IN_8$) are ‘0’. Signal probability of nodes in Fig. 2.4 is labeled as format $P_0, P_1$ with assumption that primary inputs have equal probability of values. As shown in Fig. 2.4, HT trigger signal is active (low voltage) in 1/256 rate, and passive (high voltage) in 255/256 rate, that is $P_0 \ll P_1$. Based on geometric distribution, the average number of primary input patterns that required for triggering output of 8-bit OR gate is

$$N = \frac{1}{P_d + P_a} - 1 = \frac{1}{\frac{255}{256}} - 1 = 256$$

(2.3)

where $P_p$ and $P_a$ are probability of 8-bit OR gate output being passive and active, respectively; as seen, average 256 input patterns are required to trigger the rare signal (output of 8-bit OR gate).

Fig. 2.4 8-Bit OR Gate, Probability Label ($P_0$, $P_1$)

A PIC is developed as shown in Fig. 2.5 to increase the probability of signal transition when $P_0 \ll P_1$. In Fig. 2.5, NODE is the rare signal node input; OUT is the
node output; \( TS \) (testing switch) is switch of signal transition increase, node signal transition probability is increased when \( TS \) is ‘1’; input \( SW \) (square wave) alternately provides ‘1’ and ‘0’ in a certain frequency. To turn on \( PIC \), input \( TS \) is switched to ‘1’, CMOS switch is on, then square wave propagates from \( SW \) to AND gate. The rare signal node is connected to another input (\( NODE \)) of AND gate. Thereafter \( OUT \) receives the same value as \( NODE \) when \( SW \) is ‘1’, while \( OUT \) remains ‘0’ when \( SW \) is ‘0’. Because \( SW \) is square wave and \( OUT \) is always ‘0’ when \( SW \) is ‘0’, \( OUT \) should be ‘0’ in halftime. In another halftime, \( SW \) switches to ‘1’, \( OUT \) is virtually remaining ‘1’ when \( P_0 \ll P_1 \). Eventually, \( P_1 \) and \( P_0 \) of signal \( OUT \) are much closer compared to the original design.

Based on simulation in CMOS IBM 90nm technology, power consumption of \( PIC \) in passive and active mode are 8.839nW and 367.4nW, respectively.

![Fig. 2.5 Probability Increase Circuit for \( P_0 \ll P_1 \)](image)

The 8-bit OR gate with \( PIC \) is present in Fig. 2.6, as seen in which, the probability of values in \( OUT \) is much closer (289/1024, 735/1024). The average number of primary input patterns that required to trigger signal \( OUT \) is

\[
N = \frac{1}{P_p \cdot P_a} - 1 = \frac{1}{\frac{735}{1024} \cdot \frac{289}{1024}} - 1 = 4
\]

(2.4)

As seen, only 4 rather than 256 input patterns are required to trigger the rare signal \( OUT \) (output of 8-bit OR gate).
2.2.2.2 PIC Overhead

8-bit, 16-bit and 32-bit OR gates are used as CUT simulating with/without PIC. The activation time of rare signal in CUTs is shown in Table 2.3, where CUTs operate at input frequency of 1GHz. As seen, the rare signal activation time is tremendous reduced by PIC for all CUTs with varied sizes.

Table 2.3 Activation Time of Rare Signal in CUTs

<table>
<thead>
<tr>
<th>CUT</th>
<th>CUT without PIC</th>
<th>CUT with PIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-bit AND</td>
<td>256ns</td>
<td>4ns</td>
</tr>
<tr>
<td>16-bit AND</td>
<td>66us</td>
<td>4ns</td>
</tr>
<tr>
<td>32-bit AND</td>
<td>4.3s</td>
<td>4ns</td>
</tr>
</tbody>
</table>

Overhead of PIC in CUTs is shown in Table 2.4, where circuit size is estimated by number of transistors; power is simulated in Cadence with CMOS IBM 90nm technology when PIC is off (dormant mode); delay is circuit critical path worst delay. As seen, even the parameter overhead of PIC is huge for relatively small CUT, it is
decreased linearly with the increase of corresponding CUT size. As discussed in section 2.2.1, the PIC overhead in a chip can be restricted by: 1) decrease the targeting transition probability level; 2) apply less PIC in security-robust portion of circuit compared to security-sensitive portion of circuit.

Table 2.4 Overhead of PIC for Rare Signal when $P_0 \ll P_1$

<table>
<thead>
<tr>
<th>Size</th>
<th>CUT</th>
<th>CUT without PIC</th>
<th>CUT with PIC</th>
<th>Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-bit OR</td>
<td>42</td>
<td>62</td>
<td>47.6%</td>
<td></td>
</tr>
<tr>
<td>16-bit OR</td>
<td>90</td>
<td>106</td>
<td>17.8%</td>
<td></td>
</tr>
<tr>
<td>32-bit OR</td>
<td>186</td>
<td>202</td>
<td>8.6%</td>
<td></td>
</tr>
<tr>
<td>Power (μW)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8-bit OR</td>
<td>3.064</td>
<td>3.727</td>
<td>21.6%</td>
<td></td>
</tr>
<tr>
<td>16-bit OR</td>
<td>6.248</td>
<td>6.446</td>
<td>3.2%</td>
<td></td>
</tr>
<tr>
<td>32-bit OR</td>
<td>12.61</td>
<td>12.81</td>
<td>1.6%</td>
<td></td>
</tr>
<tr>
<td>Delay (ps)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8-bit OR</td>
<td>137</td>
<td>173</td>
<td>26.3%</td>
<td></td>
</tr>
<tr>
<td>16-bit OR</td>
<td>180</td>
<td>216</td>
<td>20%</td>
<td></td>
</tr>
<tr>
<td>32-bit OR</td>
<td>260</td>
<td>290</td>
<td>11.5%</td>
<td></td>
</tr>
</tbody>
</table>

2.3 PIC Insertion Algorithm

Circuit idle nodes remains in a signal status (either high or low) for most lifetime, while leave another signal status as rare signal. The proposed PIC increases signal transaction probability by using logic gates to generate rare signal in halftime. The procedure of PIC insertion is shown in Fig. 2.7.

1. Calculate $P_0$ and $P_1$ for all nodes on chip based on input probabilities (assume unknown input probabilities are equal, $P_0 = P_1$). Order nodes by transition probability, which is equal to the product of $P_0$ and $P_1$. 

24
2. Select transition probability threshold \( (TP_t) \). All nodes with transition probability less than \( TP_t \) are classified in set \( A \).

3. If set \( A \) is empty, all nodes with transition probability greater than threshold, the process is done; otherwise, continue to next step.

4. Typically, rare situation accumulates by unbalanced signal probability one level by one level from primary inputs to outputs, so nodes with less transition probability may relatively after nodes with more transition probability. Increasing transition probability of nodes closer to primary inputs will somehow benefit transition probability of succeeding nodes. Therefore, PIC is inserted to the first node (with maximum transition probability) in set \( A \). Go back to step 1.

---

**Fig. 2.7 PIC Insertion Flow**

### 2.4 Experimental Result

A 64-bit binary comparator, shown in Fig. 2.8, is applied as CUT to evaluate the effectiveness of the PIC insertion flow. Binary comparator is basic digital arithmetic component that operates to compare two binary numbers. A 64-bit binary comparator
has two 64-bit binary input ($A_{63}-A_0 \& B_{63}-B_0$) and three binary outputs, which indicate $A>B$, $A<B$, or $A=B$ [35].

Fig. 2.8 64-Bit Binary Comparator

The signal probability of node is indicated as $P_{nm} = (P_0, P_1)$, in which $n$ stands for node, $m$ is node number; $P_0$ and $P_1$ are probability of node with signal ‘0’ and ‘1’, respectively. With assumption that primary inputs have $P_0 = P_1$, signal probability of nodes in original 64-bit binary comparator are

\[
P_{n_1} = P_{n_2} = \left(\frac{3}{8}, \frac{5}{8}\right)
\]

\[
P_{n_3} = \left(\frac{1}{4}, \frac{3}{4}\right)
\]

\[
P_{n_4} = P_{n_5} = \left(\frac{1}{2}, \frac{1}{2}\right)
\]

\[
P_{n_6} = \left(\frac{1}{2^{248}}, \frac{2^{248}-1}{2^{248}}\right)
\]

\[
P_{n_7} = P_{n_8} = \left(\frac{1}{2}, \frac{1}{2}\right)
\]
\[ P_{n_9} = \left( \frac{2^{32}}{2^{32}}, \frac{2^{32} - 2^{32}}{2^{32}} \right) \]

\[ P_{n_{10}} = P_{n_{11}} = \left( \frac{1}{2}, \frac{1}{2} \right) \]

\[ P_{n_{12}} = \left( \frac{2^{64}}{2^{64}}, \frac{2^{64} - 2^{64}}{2^{64}} \right) \]

Signal transition probability (TP) is equal to the product of \( P_0 \) and \( P_1 \). Then TPs in CUT are as following,

\[ TP_{n_1} = TP_{n_2} = \frac{3}{8} * \frac{5}{8} \approx 0.23 \]

\[ TP_{n_3} = \frac{1}{4} * \frac{3}{4} \approx 0.19 \]

\[ TP_{n_4} = TP_{n_5} = \frac{1}{2} * \frac{1}{2} \approx 0.25 \]

\[ TP_{n_6} = \frac{1}{2^{248}} * \frac{2^{248} - 1}{2^{248}} \approx 0 \]

\[ TP_{n_7} = TP_{n_8} = \frac{1}{2} * \frac{1}{2} \approx 0.25 \]

\[ TP_{n_9} = \frac{2^{32}}{2^{32}} * \frac{2^{32} - 2^{32}}{2^{32}} \approx 0 \]

\[ TP_{n_{10}} = TP_{n_{11}} = \frac{1}{2} * \frac{1}{2} \approx 0.25 \]

\[ TP_{n_{12}} = \frac{2^{64}}{2^{64}} * \frac{2^{64} - 2^{64}}{2^{64}} \approx 0 \]

The diminishing order of TP is \( TP_{n_4}, TP_{n_5}, TP_{n_7}, TP_{n_8}, TP_{n_{10}}, TP_{n_{11}}, TP_{n_1}, TP_{n_2}, TP_{n_3}, TP_{n_6}, TP_{n_9}, TP_{n_{12}} \).

We set 0.1 as signal transition probability threshold (TP\(_t\)) in this example. Then node \( n_6, n_9, \) and \( n_{12} \) are classified to set \( A \) \( (A = \{n_6, n_9, n_{12}\}) \). According to algorithm step 4, PIC is embedded in node \( n_6 \) and other “\( A=B \)” signals in stage 2. This circuit is denoted as 64-bit binary comparator with 1-level PIC.

Re-calculate signal probability and signal transition probability of nodes in 64-bit binary comparator with 1-level PIC. The updated results are as:
\[ P_{n_1} = P_{n_2} = \left( \frac{3}{8}, \frac{5}{8} \right) \]

\[ P_{n_3} = \left( \frac{1}{4}, \frac{3}{4} \right) \]

\[ P_{n_4} = P_{n_5} = \left( \frac{1}{2}, \frac{1}{2} \right) \]

\[ P_{n_6} = \left( \frac{1}{2}, \frac{1}{2} \right) \]

\[ P_{n_7} = P_{n_8} = \left( \frac{1}{2}, \frac{1}{2} \right) \]

\[ P_{n_9} = \left( \frac{1}{16}, \frac{15}{16} \right) \]

\[ P_{n_{10}} = P_{n_{11}} = \left( \frac{1}{2}, \frac{1}{2} \right) \]

\[ P_{n_{12}} = \left( \frac{1}{256}, \frac{255}{256} \right) \]

\[ TP_{n_1} = TP_{n_2} = \frac{3}{8} * \frac{5}{8} \approx 0.23 \]

\[ TP_{n_3} = \frac{1}{4} * \frac{3}{4} \approx 0.19 \]

\[ TP_{n_4} = TP_{n_5} = \frac{1}{2} * \frac{1}{2} \approx 0.25 \]

\[ TP_{n_6} = \frac{1}{2} * \frac{1}{2} \approx 0.25 \]

\[ TP_{n_7} = TP_{n_8} = \frac{1}{2} * \frac{1}{2} \approx 0.25 \]

\[ TP_{n_9} = \frac{1}{16} * \frac{15}{16} \approx 0.06 \]

\[ TP_{n_{10}} = TP_{n_{11}} = \frac{1}{2} * \frac{1}{2} \approx 0.25 \]

\[ TP_{n_{12}} = \frac{1}{256} * \frac{255}{256} \approx 0 \]

Then, the updated set \( A \) consists of node \( n_9 \) and \( n_{12} \). PIC is embedded in node \( n_9 \) and other “\( A = B \)” signals in stage 3. This modified circuit is named 64-bit binary comparator with 2-level PIC.

Repeat calculation for signal probability and signal transition probability of nodes in 64-bit binary comparator with 2-level PIC. The updated results are as:
\[ P_{n_1} = P_{n_2} = \left( \frac{3}{8}, \frac{5}{8} \right) \]

\[ P_{n_3} = \left( \frac{1}{4}, \frac{3}{4} \right) \]

\[ P_{n_4} = P_{n_5} = \left( \frac{1}{2}, \frac{1}{2} \right) \]

\[ P_{n_6} = \left( \frac{1}{2}, \frac{1}{2} \right) \]

\[ P_{n_7} = P_{n_8} = \left( \frac{1}{2}, \frac{1}{2} \right) \]

\[ P_{n_9} = \left( \frac{17}{32}, \frac{15}{32} \right) \]

\[ P_{n_{10}} = P_{n_{11}} = \left( \frac{1}{2}, \frac{1}{2} \right) \]

\[ P_{n_{12}} = \left( \frac{289}{1024}, \frac{735}{1024} \right) \]

\[ TP_{n_1} = TP_{n_2} = \frac{3}{8} \cdot \frac{5}{8} \approx 0.23 \]

\[ TP_{n_3} = \frac{1}{4} \cdot \frac{3}{4} \approx 0.19 \]

\[ TP_{n_4} = TP_{n_5} = \frac{1}{2} \cdot \frac{1}{2} \approx 0.25 \]

\[ TP_{n_6} = \frac{1}{2} \cdot \frac{1}{2} \approx 0.25 \]

\[ TP_{n_7} = TP_{n_8} = \frac{1}{2} \cdot \frac{1}{2} \approx 0.25 \]

\[ TP_{n_9} = \frac{17}{32} \cdot \frac{15}{32} \approx 0.25 \]

\[ TP_{n_{10}} = TP_{n_{11}} = \frac{1}{2} \cdot \frac{1}{2} \approx 0.25 \]

\[ TP_{n_{12}} = \frac{289}{1024} \cdot \frac{735}{1024} \approx 0.2 \]

All updated \( TP \)s are greater than \( TP_t \) (0.1). The PIC insertion process is done.

The node with minimum signal transition probability in CUT without PIC and CUT with 2-level PIC is \( n_{12} \) and \( n_3 \), respectively. Assume that the rare signal in node with minimum transition probability is used as HT trigger signal. Then, the average number of primary input patterns that required to trigger HT is
\[ N = \frac{1}{TP_{n_{12}}} - 1 = \frac{1}{2^{64} - 2^{64} - 2^{64}} - 1 \approx 1.8 \times 10^{19} \]

for CUT without PIC, and

\[ N = \frac{1}{TP_{n_{3}}} - 1 = \frac{1}{4^{4}} - 1 \approx 4 \]

for CUT with 2-level PIC. Simulation results are listed in Table 2.5, where CUT is 64-bit binary comparator; delay is circuit critical path worst delay; power is average power consumption when PIC is off (in dormant mode); circuit size is estimated by number of transistors; time stands for average time to active HT when input signal is at frequency of 1GHz.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>CUT</th>
<th>CUT with 1-level PI</th>
<th>CUT with 2-level PI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Parameter</td>
<td>Overhead</td>
</tr>
<tr>
<td>Delay (ps)</td>
<td>712</td>
<td>840</td>
<td>18%</td>
</tr>
<tr>
<td>Power (mW)</td>
<td>12.78</td>
<td>12.78</td>
<td>0%</td>
</tr>
<tr>
<td>Size</td>
<td>1314</td>
<td>1434</td>
<td>1.5%</td>
</tr>
<tr>
<td>Time</td>
<td>570 years</td>
<td>256 ns</td>
<td>100%</td>
</tr>
</tbody>
</table>

As shown in Table 2.5, CUT with 1-level PI has the same improvement on HT activation time compared to CUT with 2-level PI, however consumes almost half delay overhead. Then 64-bit binary comparator with 1-level PI is chosen as the final modified circuit. As seen, for the id lest node in original CUT that may take 570 years to active HT, the PIC insertion can even reduce the activation time to 256ns. It can be claimed that the proposed PIC has ability to modify all nodes in CUT having an acceptable signal transition probability.
2.5 Conclusion

A novel design (PIC) is introduced to increase transition probability of idle node on chip that may be used as HT trigger signal. It may benefit HT detection by reducing the HT activation time. 64-bit binary comparator is used to evaluate the effectiveness of the PIC. Trigger time of rare signal in CUT is tremendously reduce with modest circuit overhead.
3 Timing Analysis-based Hardware Trojan Detection

A timing analysis-based microelectronic circuit Hardware Trojan (HT) detection methodology is proposed in this chapter. The detection circuit can be adopted in combinational and sequential microelectronic circuits. The proposed technique is implemented in IBM 90nm CMOS process and Xilinx ISE FPGA. Based on experimental results, with one detection circuit embedded in testing-path, a HT with size that is 2.81% of host-circuit size (15.63% of testing-path size) is detectable at detection probability of 90% with a 10% probability of a false positives. Both detectable HT size and detection probability can be improved by adding more detection circuits to testing-path. The probability of false positives is controlled by the test clock period. The discussion in this chapter is substantially drawn from [36][37], where we first reported the development and evaluation of this technique.

3.1 Introduction

Recently, HT detection based on timing analysis has arisen as a popular technique. Embedded HT will add extra capacitance, resulting in more charging and discharging delays to Trojan affected paths. HT can be revealed by measuring the differential timing characteristics of the attacked circuit.

Most publications based on timing analysis techniques focus on algorithms to distinguish the timing difference caused by HT and methods to augment this difference. Authors in [26] chose many clean (free of HTs) ICs and run a variety of input patterns to record specific path delays establishing key delay fingerprints. Afterwards, circuit-under-test (CUT) is tested under the same input patterns and verified by comparing timing parameters with genuine fingerprint. Due to the complexity of millions of paths in a complicated chip and instability of golden standards caused by inter and intra-die variations, this method becomes infeasible for large circuit designs.
Instead of comparing with genuine fingerprint, authors in [38] compare the delay of testing-paths with a set of symmetric paths, which are other paths that have the same topology as the testing-paths. The delay of testing-paths and symmetric paths will be the same unless an inserted HT breaks symmetry. Symmetry can naturally exist in ICs or be artificially added. This method can avoid the difficulty and cost in finding a golden model for all ICs with variable parameters. However, the detection accuracy suffers from intra-die parameter variation, and the detection method is limited in that huge area-overhead may emerge due to a bulk of artificial symmetries are required. Also, massive effort is accumulated in finding existing symmetric paths and the specific test vectors to measure desired path delays.

Authors in [39] increase the system clock frequency to create clock glitches until resulting in faulty operation. Then the delay of IC critical paths for several bits are estimated with the faulted outputs and the corresponding clock frequency. The simulated path delays are compared to golden parameters to ascertain security reliability. Statistical analysis method is introduced in [40] to facilitate the identification of HT. A test-vector selection scheme and a novel timing measurement structure proposed in [41] are effective in accurate path-delay measurement.

In [42], a HT detection method based on physical unclonable function (PUF) is proposed, as shown in Fig. 3.1. It is used to detect HT in register-to-register paths. The registers (FF1, FF2) in main circuit are triggered by the main system clock (clk1). HT detection circuitry is within the dotted box, and the register in it (FF3) is triggered by a shadow clock (clk2), which has the same frequency with main system clock (clk1) and a controlled negative shift. The negative shift of shadow clock makes FF3 to be triggered earlier than FF2, thereby output of combinational circuit arrives comparator through FF3 ahead of it through FF2. Then the shadow clock negative shift is increased.
until the register outputs are unequal. That clock shift time is claimed to be the \textit{combinational circuit} delay. The \textit{combinational circuit} is suspicious of being HT-attacked if the measured delay is substantially different from the pre-determined designed timing. This technique is at-speed detection, which can be applied at both test-time and run-time, but it requires extra circuit with large over-head to control skewed clock (\textit{clk}2). Moreover, it can only be used in sequential circuit register-to-register paths.

![Fig. 3.1 Block Diagram of HT Detection Circuit Using PUF](image)

A modified timing analysis-based HT detection technique is proposed in this chapter, in which the clock skew control circuit is eliminated to simplify the detection circuit. The experimental results show that the HT detection circuit overhead is competitive compared to state-of-the-art with similar detection probability. The main contributions of this technique are as follows:

- HT detection circuit area, timing, and power overhead on host-circuit are reduced.

  The proposed detection circuit operates with main system clock, so the specific clock skew control circuit (normally with thousands of gates) for HT detection is eliminated.

- The proposed HT detection technique is not restricted to be applications with register-to-register paths (e.g., [39][42]). This technique can be used on any circuit path by isolating the path with extra registers.
• Location of HT can be estimated. The detection signals of each HT testing-path are read out in series. The path of suspected HT can be determined by the location of abnormal detection signal.

• Tunable detection probability to accommodate different security requirements of circuit designs. HT detection probability can be increased by more detection circuits pairing with one testing-path, meanwhile detection circuit overhead is increased. The selection of testing-path paired with one detection circuit is determined by chip designer based on vulnerability of the path, desired HT resistance, and limitations on parameter overhead. For example, the security-sensitive portion of circuit (e.g., memory) is required to be covered by more detection circuits to achieve more accurate HT detection. Memory circuit is security-sensitive, because the key data stored in memory could be a target for attackers. While security-robust portion of circuit (e.g., critical path) can be covered by less (or even zero) detection circuits to reduce workload and circuit overhead. Critical path is security-robust, because any extra connection on critical path will slow chip operating frequency, that can be easily revealed in post-fab functional test.

3.2 Timing Analysis-based HT Detection

This section introduces the timing analysis-based HT detection algorithm, functional block diagram and a sub-circuit design. To simplify the introduction of detection algorithm, all cases in this section assume no manufacturing or environmental variations, and experimental results are based on schematic simulation with tt (typical-typical) corner. The effect of operating variations is considered in section 3.3.
3.2.1 Timing Analysis-based HT Detection Algorithm

In order to avoid detection, typically HTs are designed staying dormant until activation. The proposed detection technique algorithm and implementation are with premise that the HT remains dormant (sleeping mode) during detection process.

An HT circuit embedded in testing-path will add extra capacitance, resulting in more charging and discharging delays to the testing-path [43][44]. The delay time (time from applying an input pattern to signal arriving at output) will be increased for a HT-attacked CUT compared to HT-free CUT due to the additional delay caused by HT circuit. The input and output signals of testing-path with/without Trojan are shown in Fig. 3.2, where \( t_c \) is the designed delay of the testing-path. The proposed detection method compares output signal at the calibrated CUT delay time \( t_c \) and at double of the calibrated CUT delay time \( 2t_c \) to facilitate HT detection. For ideal case, signal arrives output at time \( t_c \) in HT-free CUT, and the output signal remains fixed until time \( 2t_c \) with assumption that input is unchanged, so the output at time \( t_c \) and \( 2t_c \) are equal. In HT-attacked CUT, due to the increased delay caused by HT, signal arrives at the output after a time greater than \( t_c \), so the output at time \( t_c \) and \( 2t_c \) are unequal.

![Fig. 3.2 Signals in Trojan-Free (TF) and Trojan-Attacked (TA) Testing-Path](image)

The flowchart of timing analysis-based HT detection algorithm consists of five steps, as shown in Fig. 3.3. Based on circuit simulation, the delay time of Trojan-free CUT testing-path is measured as \( D \), which is used as golden reference (calibrated delay time). Then input of testing-path is scanned in, and output at time \( D \) and \( 2D \) is captured
as signal $O_1$ and $O_2$, respectively. If signal $O_1$ and $O_2$ are same, the CUT is Trojan-free; otherwise, the CUT is Trojan-attacked. Because of manufacturing and environmental variations, the delay time of fabricated CUTs will have a statistical deviation from the ideal calibrated delay time of testing-path. Therefore, detection probability suffers from manufacturing and environmental variations, that must be considered in the implementation of HT detection. The implementation result of manufacturing and environmental-aware HT detection is introduced in section 3.4.

![Timing Analysis-based HT Detection Algorithm](image)

Fig. 3.3 Timing Analysis-based HT Detection Algorithm

Furthermore, a ring-oscillator can be introduced to estimate operation temperature and process variation of individual testing-path in CUT, so that the golden reference time $D$ can be calibrated for each testing-path to benefit HT detection probability. The use of ring-oscillator in the detection process will be discussed in section 3.3.2.

### 3.2.2 Timing Analysis-based HT Detection Block Diagram

As shown in Fig. 3.4, the proposed HT detection block diagram includes CUT
block (testing-path) and Detection Circuit block. Detection Circuit block represents HT detection circuit, consisting of flip-flops (FF1 & FF2), a pass transistor Switch, XOR gate, and feedback-AND-OR gate (FAO). The specification and design of FAO are introduced in section 3.2.3.

Fig. 3.4 HT Detection Block Diagram

For HT detection in combinational circuit (testing-path 1 in Fig. 3.5), flip-flops (FF1 & FF2) are added to extract testing-path for further measurement. For HT detection in testing-path adjacent to flip-flop in sequential circuit (testing-path 2 & 3 in Fig. 3.5), detection circuit may share flip-flop with CUT. For example, detection circuit on testing-path 2 in Fig. 3.5 can use FF3 in CUT as FF1, while detection circuit on testing-path 3 can use FF4 and FF5 in CUT as FF1 and FF2. Compared to the timing analysis-based design in [42], overhead of the proposed HT detection circuit is benefited from removing skewed clock (clk2 in Fig. 3.1), which requires 6469 gates for clock control circuit. The main system clock is shared by all detection circuits in CUT. To work functionally, clock frequency in detection circuit is dependent on the corresponding testing-path. So only one testing-path is detected in one clock cycle with a specific clock frequency. The size of a testing-path is proportional to the size of
detectable HT in the testing-path. More HT detection circuits applied on one testing-path will increase detection probability, meanwhile resulting in more circuit over-head. The selection and size of testing path are determined by chip designer and testing engineer based on HT resistance to detection, the required HT detection probability, and limitations on parameter overhead.

![Diagram](image)

**Fig. 3.5 Testing-Path in CUT**

The timing details of input and output waveforms for the proposed HT detection applying on circuit in Fig. 3.4 is shown in Fig. 3.6. Signal propagation time of a testing-path is measured as time $D$. The off-chip system clock period is set up as $2D$. As seen in Fig. 3.6, when detection starts, a rising edge input signal is scanned in; at next clock positive edge (labelled as time $t_1$), output of $FF1$ (node $n1$) captures the input rising. The remaining circuit operations consider two cases: 1) Trojan-free, and 2) Trojan-attacked.

1) Trojan-free circuit

In Trojan-free circuit, as shown in Fig. 3.6 (a), both $n3$ and $O_1$ are pulled up to ‘1’ at time $D$ after input changes. $FAO$ gate initializes when inputs are both ‘1’ or ‘0’, then functions as either an OR gate when output is ‘1’ or an AND gate when output is ‘0’. The implementation of $FAO$ is introduced in next section. Afterwards, at the clock falling edge (labelled as time $t_2$), two inputs of $FAO$ ($\bar{clk}$ & $n1$) are both ‘1’, so output of $FAO$ raises to ‘1’ and signal $n2$ falls to ‘0’ turn off $SWITCH$. The detection circuit is designed so that signal propagation from $n3$ to $O_1$ is faster than that from $clk$ to $n2$, then $O_1$ rises before turning off $SWITCH$. Because $FAO$
gate functions as an OR gate when output is ‘1’, if one input of $FAO (n1)$ remains ‘1’, the output of $FAO$ gate will be ‘1’ regardless of another input ($\overline{clk}$), eventually $SWITCH$ remains off. Signal $n3$ transfers to $O_2$ through $FF2$ at next clock rising edge (labelled as time $t3$). The detection result, which is equal to $O_1 \oplus O_2$, remains low after time $t3$.

Fig. 3.6 Operation Waveform of HT Detection in Corner $TT$ (typical-typical), (a)

Trojan-Free (TF), (b) Trojan-Attacked (TA)
2) Trojan-attacked circuit

In Trojan-attacked circuit, extra time is required to charge the HT capacitor that is interfaced to testing-path. As shown in Fig. 3.6 (b), signal n3 is pulled up to ‘1’ at time $t_2'$ which is later than $t_2$. Because SWITCH is turned off at time $t_2$, signal $O_1$ cannot capture $n3$ changes, and stays low. Signal $O_2$ captures $n3$ through $FF2$ at the succeeding clock rising edge (time $t_3$). The detection result is pulled to ‘1’ at time $t_3$.

After HT detection takes place, the detection result is read out by a scan-chain that connects to detection result of all detection circuits in CUT. Then off-chip system clock frequency is adjusted for next testing-path. This method can detect one testing-path with one clock period.

The testing-path delay times will vary due to manufacturing and environmental variations. The effect of these variations will be considered in section 3.3 with the introduction of a probability of detecting an HT.

### 3.2.3 FAO Gate Implementation

FAO gate consists of two inputs ($A$ & $B$), one output ($\overline{Y}$), and it can be initiated by applying inputs both ‘1’ or both ‘0’, then output $\overline{Y}$ is ‘1’ or ‘0’, respectively. As design specification, the function of FAO gate after initiation is represented by (3.1)

$$
\overline{Y} = \begin{cases} 
A \cdot B, & \text{when } \overline{Y} = 0 \\
A + B, & \text{when } \overline{Y} = 1 
\end{cases} \tag{3.1}
$$

FAO transistor level design and paths operation are shown in Fig. 3.7 and Table 3.1, respectively. Output $\overline{Y}$ is uncertain before initiation. When inputs are “11”, pull-down-path $T7, T8$ is on, output $\overline{Y}$ is initiated to ‘1’; when inputs are “00”, pull-up-path $T3, T2, T1/T4$ is on, output $\overline{Y}$ is initiated to ‘0’. As seen in Table 3.1, FAO functions as either an AND gate when ‘$\overline{Y}$’ is ‘0’, or an OR gate when ‘$\overline{Y}$’ is ‘1’. This design can work as AND or OR gate that is dependent on circuit output signal, therefore the
The proposed design is denoted as feedback-AND-OR gate.

Fig. 3.7 Feedback-AND-OR (FAO) Gate

Table 3.1 Paths Operation in FAO Gate

<table>
<thead>
<tr>
<th>Current Output</th>
<th>Input</th>
<th>Next Output</th>
<th>Path On</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\overline{\gamma}$</td>
<td>A</td>
<td>B</td>
<td>$\gamma$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\overline{\gamma}$</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>$\overline{\gamma}$</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>$\overline{\gamma}$</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>$\overline{\gamma}$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>$\gamma$</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>$\gamma$</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>$\gamma$</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>$\gamma$</td>
</tr>
</tbody>
</table>
3.2.4 Deployment of Timing Analysis-based HT Detection

There are several ways to categorize HT, such as categorizations based on physical, activation, action, location characterization [4]. This section introduces HT categorization based on location of HT, which can significantly vary the effect of HT on host-circuit timing parameters.

![Diagram](a)

![Diagram](b)

![Diagram](c)

Fig. 3.8 HT Location in CUT, (a) In-Path, (b) By-Path, (c) Off-Path

Based on location characteristics of Trojan in host-circuit, Trojan is partitioned into three categories: in-path, by-path, and off-path, as shown in Fig. 3.8. When input pattern is applied, the location of HT plays an important role on the ability to detect it. The propagation signal in CUT will go through the extra paths caused by in-path HT, which affects the measured transient time for some extend. The chip’s function can be changed by an in-path HT through adding additional logic or bypassing the original logic. A by-path HT affects the CUT’s original path by capacitive load introduced by the HT, which affects the measured transient time for much less extend. A by-path HT may be used to transit IC internal information to the attacker. Off-path HT is isolated from the original circuit path, resulting in negligible timing alteration on original signal.
propagation. An off-path HT may be used to tremendously change circuit temperature, eventually modify the chip’s parametric properties, such as delay and power [5]. Although by-path and off-path HT has relatively tiny effect on path timing parameter, HT’s control and action logic add extra Trojan circuit to the host-chip that will consume extra power compared to original design. The detection of this type of Trojan is under study based on combinational of power and timing-analysis in chapter 5. The proposed method in this chapter is introduced for in-path and by-path HT detection.

3.3 Schematic Implementation

The HT detection circuit is implemented in Cadence Schematic with IBM 90nm CMOS process. Monte Carlo analysis is used to account for operating variations.

The proposed HT detection method is implemented on 4, 8, 16-bit ripple-carry-adders (RCAs) with in-path and by-path HTs, respectively. Signal n1 and n3 in Fig. 3.4 are carry-in (C_in) and carry-out (C_out) of RCA, respectively. Input vector of CUT is set as C_out rises with C_in rising.

3.3.1 Operating Variation-Aware HT Detection

For ideal case, the circuit delay time will be increased (greater than t_c) for a HT-attacked CUT compared to HT-free CUT due to the additional delay caused by HT circuit, shown in Fig. 3.2, so the output at time t_c and 2t_c are unequal. However, in actual case, because of manufacturing process variation, circuit delay time varies in a range, as shown in Fig. 3.9, in which the actual circuit delay varies in the range labelled as from min to max. Increasing circuit calibrated delay time t_c will cover more delay cases in Trojan-free case that decreases HT detection false positive (FP), meanwhile increased t_c may miss more delays in Trojan-attacked case that decreases HT detection probability. Also, as shown in Table 3.2, decreased t_c will increase both
detection FP and detection probability. Note that false positive is the fault that asserts Trojan-attacked for a Trojan-free circuit (false alarm).

![Diagram showing ideal signals in Trojan-Free (TF) and Trojan-Attacked (TA) Testing Path](image)

Fig. 3.9 Ideal Signals in Trojan-Free (TF) and Trojan-Attacked (TA) Testing Path

<table>
<thead>
<tr>
<th>$t_c$</th>
<th>False Positive</th>
<th>Detection Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Increased</td>
<td>Decreased</td>
<td>Decreased</td>
</tr>
<tr>
<td>Decreased</td>
<td>Increased</td>
<td>Increased</td>
</tr>
</tbody>
</table>

Table 3.2 HT Detection Parameters with Variable Calibrated Delay Time

To find the relationship of circuit fan-out and circuit timing alteration caused by operating variation, an experimentation is set up as follows. Ten inverters connected in serial, shown in Fig. 3.10 (a), is used as CUT with fan-out $I$, while extra number of $n-I$ inverters connected to each interval node, shown in Fig. 3.10 (b), is used as CUT with fan-out $n$. The experimental results are listed in Table 3.3, in which ideal delay is based on schematic simulation with $tt$ (typical-typical) corner; actual delay is based on schematic simulation with Monte Carlo analysis (100 iterations) to account for random process variation and mismatch. As seen, the actual delay variation is decreased with number of circuit fan-out increasing. Because it is rare that all fan-outs succeeding a component pushing its process variation to limitation at the same direction. Therefore, we claim that more fan-outs in a circuit design lead to low operating parameter alteration associated with process variation, eventually benefit HT detection probability,
whereas attendant extra circuit delay.

(a)

(b)

Fig. 3.10 CUT with fan-out 1 (a), and fan-out n (b)

Table 3.3 Delay of CUT with different fan-out

<table>
<thead>
<tr>
<th>Fan-out</th>
<th>Delay (ps)</th>
<th>Delay Variation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ideal</td>
<td>Actual</td>
</tr>
<tr>
<td>1</td>
<td>133.87</td>
<td>105.84-174.01</td>
</tr>
<tr>
<td>2</td>
<td>203.87</td>
<td>158.96-249.47</td>
</tr>
<tr>
<td>3</td>
<td>275.14</td>
<td>221.92-329.87</td>
</tr>
<tr>
<td>4</td>
<td>347.87</td>
<td>284.43-408.86</td>
</tr>
<tr>
<td>5</td>
<td>421.02</td>
<td>355.48-489.70</td>
</tr>
</tbody>
</table>

To prove that circuit with more fan-outs benefit HT detection probability, the proposed timing analysis-based HT detection method is applied on a CUT (16-bit ripple-carry-adder) with different number of fan-outs. Inverters are used connecting to intermediate nodes to account for extra fan-outs. The experimental results are shown in Table 3.4, in which HT detection is at detection probability of 90% with a 10% probability of a false positives; the sum of widths of all NMOS and PMOS in circuit is used to estimate circuit size. As seen, circuit with increased number of fan-outs possess lower delay variation and ability to detect HT with smaller size that means detection probability is increased.
Table 3.4 HT detection for CUT with different number of fan-outs

<table>
<thead>
<tr>
<th>Fan-out</th>
<th>Host-Circuit Size (μm)</th>
<th>Actual Delay Variation</th>
<th>HT/Host-Circuit Size Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>170</td>
<td>28.73%</td>
<td>2.81%</td>
</tr>
<tr>
<td>2</td>
<td>173.84</td>
<td>27.62%</td>
<td>2.73%</td>
</tr>
<tr>
<td>3</td>
<td>177.68</td>
<td>27.15%</td>
<td>2.68%</td>
</tr>
<tr>
<td>4</td>
<td>181.52</td>
<td>26.87%</td>
<td>2.66%</td>
</tr>
<tr>
<td>5</td>
<td>185.36</td>
<td>26.51%</td>
<td>2.59%</td>
</tr>
</tbody>
</table>

3.3.2 Compensation of Manufacturing and Environmental Variations

Authors in [45] indicate that manufacturing and environmental variations can cause up to 30% circuit performance difference. Also, the increased logic density and operation frequency of modern ICs have exponentially magnified operational variations due to temperature fluctuations. The increased temperature reduces transistor switching speed, while increases transistor leakage current, that increases chip power consumption and propagation delay.

To compensate these effects of variation on HT detection, a ring-oscillator, whose frequency is temperature-dependent, is used to feed a counter, as shown in Fig. 3.11 [46]. The counter output can be read out to estimate ring-oscillator frequency. If the ring-oscillator is layout-close to a testing-path, the environmental variations of testing-path circuit can be estimated using ring-oscillator frequency. Then the calibrated delay signature of the testing-path considering environmental variations may be used to benefit HT detection probability.
This technique is overhead-effective, and benefits detection probability in individual testing-path in CUT. Due to the limitation of simulation tools, the ring-oscillator is not implemented, but is introduced as a potential method to optimize the detection probability.

3.3.3 In-Path HT Detection

HT occupying CUT path is defined as in-path HT, as shown in Fig. 3.8 (a). The HT detection probability and detectable HT size in CUT are discussed in this section.

3.3.3.1 HT/Host-Circuit Ratio

To find how the size of testing-path will affect detection probability, a single detection circuit is applied on three CUTs with different size (4, 8, 16-bit RCA). To determine the calibrated circuit delay, flip-flops are added to input and output of the RCA to find the maximum clock frequency of flip-flops while the RCA continues to work functionally. Inverter buffers are randomly embedded in CUT testing-path to account for HT. Both HT-free and HT-attacked CUTs are simulated by Monte Carlo analysis (100 iterations) to account for random process variation and mismatch. The detection results of HT-attacked CUTs are shown in Fig. 3.12.
(a) 4-Bit RCA
- False Positive=0%, $f_{\text{clock}}=1.19$GHz
- False Positive=10%, $f_{\text{clock}}=1.32$GHz
- False Positive=20%, $f_{\text{clock}}=1.35$GHz
- False Positive=30%, $f_{\text{clock}}=1.39$GHz

(b) 8-Bit RCA
- False Positive=0%, $f_{\text{clock}}=0.6$GHz
- False Positive=10%, $f_{\text{clock}}=0.65$GHz
- False Positive=20%, $f_{\text{clock}}=0.67$GHz
- False Positive=30%, $f_{\text{clock}}=0.69$GHz
Fig. 3.12 Detection of RCAs with In-Path HT, (a) 4-Bit, (b) 8-Bit, (c) 16-Bit RCA

*False positive* (FP) in Fig. 3.12 is the fault that asserts Trojan-attacked for a Trojan-free circuit (false alarm). The probability of a FP in a Trojan-free circuit is largely controlled by \( f_{\text{clock}} \). Increasing clock frequency constrains signal propagation time from input to output of a test circuit. As clock frequency increases, measuring the test circuit output at half the clock period and one clock period is more likely to produce conflicting results thereby increasing the probability of a FP. Each test result is based on averaging 100 Monte Carlo run and it turns out that probability of FP is essentially constant for a given clock frequency. As seen in Fig. 3.12 (a) the probability of a FP is 0% if the clock frequency is equal to or less than 1.19GHz regardless of the HT size.

As clock frequency increases, the probability of FP increases, resulting in the plots of the FP probability of 0, 10%, 20%, and 30%. On the other hand, a *false negative* (FN) (100%-detection probability) is the fault that asserts Trojan-free for a Trojan-attacked circuit. For a given clock frequency the probability of a FN is dependent on the relative size of the HT compared to the test circuit size. As expected, the probability of a FN
decreases with HT size that is supported by the results in Fig. 3.12 where each result is an average of 100 Monte Carlo runs. Each contour in Fig. 3.12 represents the probability of a FN given a fixed FP. It is noted that 100%-FN% is the probability of detecting the HT given a FP. For example, a 90% probability of detecting a HT given a 10% probability of a FP (asserting an HT when none is present) corresponds to FN and FP being equal to 10%. The data in Table 3.5 has been extracted from the Fig. 3.12 plots for equal values of FN and FP noting that 100%-FN% is the detection probability for a given FP. The “Area overhead” in Table 3.5 is the size ratio of detection circuit over CUT. The same detection circuit is applied to all three RCAs (input to output of 4, 8, or 16-bit RCA), but the total size of the RCA essentially doubles from 4 to 8 to 16-bits, so the area overhead decreases with the number of bits. Longer signal propagation path requires more transfer time, so the clock frequency for a given FP decreases and the HT size for a given detection probability increases with the number of bits in RCA.

Table 3.5 In-Path HT Detection in RCAs

<table>
<thead>
<tr>
<th>CUT</th>
<th>Area Overhead</th>
<th>Detection Probability (100%-FN)</th>
<th>FP</th>
<th>$f_{clock}$ (GHz)</th>
<th>HT Size ($\mu$m)</th>
<th>HT/Host-Circuit Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-Bit</td>
<td>14.04%</td>
<td>100%</td>
<td>0%</td>
<td>1.19</td>
<td>2.88</td>
<td>6.74%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>90%</td>
<td>10%</td>
<td>1.32</td>
<td>1.44</td>
<td>3.37%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>80%</td>
<td>20%</td>
<td>1.35</td>
<td>0.96</td>
<td>2.25%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>70%</td>
<td>30%</td>
<td>1.39</td>
<td>0.48</td>
<td>1.13%</td>
</tr>
<tr>
<td>8-Bit</td>
<td>7.03%</td>
<td>100%</td>
<td>0%</td>
<td>0.6</td>
<td>4.8</td>
<td>5.62%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>90%</td>
<td>10%</td>
<td>0.65</td>
<td>2.88</td>
<td>3.37%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>80%</td>
<td>20%</td>
<td>0.67</td>
<td>1.92</td>
<td>2.25%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>70%</td>
<td>30%</td>
<td>0.69</td>
<td>1.44</td>
<td>1.69%</td>
</tr>
<tr>
<td>16-Bit</td>
<td>3.51%</td>
<td>100%</td>
<td>0%</td>
<td>0.31</td>
<td>8.16</td>
<td>5.89%</td>
</tr>
<tr>
<td>RCA</td>
<td>90%</td>
<td>10%</td>
<td>0.33</td>
<td>4.8</td>
<td>2.81%</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-----</td>
<td>-----</td>
<td>------</td>
<td>------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>80%</td>
<td>20%</td>
<td>0.34</td>
<td>4.32</td>
<td>2.53%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>70%</td>
<td>30%</td>
<td>0.35</td>
<td>1.92</td>
<td>1.12%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To summarize, this detection technique facilitates essentially controlling the probability of FP (false alarm) with the test circuit clock frequency while the probability of detection is controlled by the relative size of HT to the test circuit size.

3.3.3.2 HT/Testing-Path Ratio

Path delay is the time from a path-input transition to a path-output transition. Path delay is caused by capacitance in the signal propagation path which cannot be charged or discharged instantaneously [47]. Signal propagates in circuit by switching the CMOS networks on and off, charging and discharging each node capacitors through the entire path. HT increases the circuit path delay by adding extra capacitance in testing-path. The extra capacitance on testing-path caused by HT is denoted as *HT on-path capacitance*. An important parameter affecting HT detection accuracy is the ratio of detectable HT on-path capacitance to testing-path total capacitance.

Gate capacitance is used to estimate the size of both HT capacitance and CUT capacitance. The capacitance at each node consists of driving load and self-load. Both loads are proportional to the transistor width [48]. Then the sum of widths of all NMOS and PMOS on testing-path is used to estimate circuit capacitance on testing-path for both CUT and HT. The detectable HT size and testing-path size at detection probability 90% is listed in Table 3.6. The *Testing-Path Size* in Table 3.6 is the sum of widths of transistors in CUT that directly interfaced to testing-path. The *HT On-Path Size* in Table 3.6 is the sum of widths of transistors in HT that directly connected to testing-path. As seen, detectable HT size is proportional to testing-path size.
Table 3.6 Detectable In-Path HT/Testing-Path Size at 90% Detection Probability

<table>
<thead>
<tr>
<th>CUT</th>
<th>Testing-Path Size (μm)</th>
<th>HT On-Path Size (μm)</th>
<th>HT/Testing-Path Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-Bit RCA</td>
<td>7.68</td>
<td>1.44</td>
<td>18.75%</td>
</tr>
<tr>
<td>8-Bit RCA</td>
<td>15.36</td>
<td>2.88</td>
<td>18.75%</td>
</tr>
<tr>
<td>16-Bit RCA</td>
<td>30.72</td>
<td>4.8</td>
<td>15.63%</td>
</tr>
</tbody>
</table>

As seen in Table 3.6, the detectable HT size ratio fluctuates in range of 15.63-18.75%. In order to provide a practical detectable HT size ratio, Linear Regression [49] is applied to find the best linear equation expressing from data in Table 3.6 and predict the detectable Trojan size with known testing path size. We use $T_0, T_1, T_2$ and $P_0, P_1, P_2$ to represent detectable HT size and testing-path size in the three RCAs above, respectively. Then Gram matrix is developed as

$$Q = \begin{bmatrix} \frac{P_0^2 + P_1^2 + P_2^2}{P_0 + P_1 + P_2} & \frac{P_0 + P_1 + P_2}{3} \\
\end{bmatrix}$$

(3.2)

$$b_0 = \begin{bmatrix} P_0 T_0 + P_1 T_1 + P_2 T_2 \\
T_0 + T_1 + T_2 
\end{bmatrix}$$

(3.3)

$$x^* = Q^{-1}b_0 = \begin{bmatrix} 0.1429 \\
0.48 \end{bmatrix}$$

(3.4)

So

$$T = [P \quad 1] \cdot x^* = 0.1429P + 0.48$$

(3.5)

where detectable HT size is $T$, detecting-path size is $P$. Equation (3.5) and the three data points in Table 3.6 used in Linear Regression are plotted in Fig. 3.13. As seen, the straight-line virtually cross origin of coordinates, so the slope of the straight line can be used to estimate the ratio of detectable HT/testing-path size. Then we can claim that a HT with size greater than 14.29% of HT-connected propagation path size can be detected by the proposed detection methodology.
3.3.4 By-Path HT Detection

A by-path HT occurs when the HT is not embedded in a CUT path but wired to a path node which is depicted in Fig. 3.8 (b). The HT detection probability and detectable HT size are discussed in this section for the case of a CUT with by-path HT.

3.3.4.1 HT/Host-Circuit Ratio

The extra propagation time caused by the by-path HT capacitive load directly connected to CUT testing-path is usually small. Instead of measuring timing parameter alteration of whole host-circuit, the proposed work measures the timing difference of specific testing-path. Then the timing difference made by HT is more obvious, and detection probability can be controlled by length of CUT testing-path paired with one detection circuit. More HT detection circuits on a testing-path will increase the
4, 8, 16-bit RCAs are used as CUT to implement by-path HT detection. Inverters are randomly connected to nodes on the testing-path to account for by-path HT load to CUT. Both Trojan-free and Trojan-attacked CUTs are simulated using Monte Carlo analysis (100 iterations) to account for random process variation and mismatch. The detection results of Trojan-attacked CUTs are shown in Fig. 3.14. As seen, with the clock frequency corresponding to the same FP, HT detection probability is proportional to Trojan size. The Trojan size is in term of the sum of widths of all NMOS and PMOS transistors in HT that directly interfaced to the CUT. CUT with larger size requires Trojan with larger size to achieve the same detection probability. The data in Table 3.7 has been extracted from the Fig. 3.14 plots for equal values of FN and FP noting that 100%-FN is the detection probability for a given FP.
Fig. 3.14 Detection of RCAs with By-Path HT, (a) 4-Bit, (b) 8-Bit, (c) 16-Bit RCA
### 3.3.4.2 HT/Testing-Path Ratio

As aforementioned, sum of transistor widths is used to estimate circuit size in this chapter. The testing-path size and the minimum detectable HT size at detection probability 90% are listed in Table 3.8. As seen, detectable HT size is proportional to testing-path size. Note that, the implementation result has potential to be improved by ring-oscillator. The HT/testing-path ratio in Table 3.8 is greater than that in the in-path HT detection in Table 3.6. By-path HT is harder to detect compared to in-path HT.

<table>
<thead>
<tr>
<th>CUT</th>
<th>Detection Probability (100%-FN)</th>
<th>FP</th>
<th>HT Size (μm)</th>
<th>HT/Host-Circuit Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-Bit RCA</td>
<td>100%</td>
<td>0%</td>
<td>3.6</td>
<td>8.42%</td>
</tr>
<tr>
<td></td>
<td>90%</td>
<td>10%</td>
<td>1.68</td>
<td>3.93%</td>
</tr>
<tr>
<td></td>
<td>80%</td>
<td>20%</td>
<td>1.2</td>
<td>2.81%</td>
</tr>
<tr>
<td></td>
<td>70%</td>
<td>30%</td>
<td>0.72</td>
<td>1.68%</td>
</tr>
<tr>
<td>8-Bit RCA</td>
<td>100%</td>
<td>0%</td>
<td>6.24</td>
<td>7.30%</td>
</tr>
<tr>
<td></td>
<td>90%</td>
<td>10%</td>
<td>3.84</td>
<td>4.49%</td>
</tr>
<tr>
<td></td>
<td>80%</td>
<td>20%</td>
<td>2.88</td>
<td>3.37%</td>
</tr>
<tr>
<td></td>
<td>70%</td>
<td>30%</td>
<td>1.68</td>
<td>1.97%</td>
</tr>
<tr>
<td>16-Bit RCA</td>
<td>100%</td>
<td>0%</td>
<td>9.84</td>
<td>7.68%</td>
</tr>
<tr>
<td></td>
<td>90%</td>
<td>10%</td>
<td>5.76</td>
<td>4.49%</td>
</tr>
<tr>
<td></td>
<td>80%</td>
<td>20%</td>
<td>4.56</td>
<td>3.56%</td>
</tr>
<tr>
<td></td>
<td>70%</td>
<td>30%</td>
<td>2.4</td>
<td>1.87%</td>
</tr>
</tbody>
</table>
Table 3.8 Detectable By-Path HT/Testing-Path Size at 90% Detection Probability

<table>
<thead>
<tr>
<th>CUT</th>
<th>Testing-Path Size (μm)</th>
<th>HT On-Path Size (μm)</th>
<th>HT/Testing-Path Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-Bit RCA</td>
<td>7.68</td>
<td>1.68</td>
<td>21.9%</td>
</tr>
<tr>
<td>8-Bit RCA</td>
<td>15.36</td>
<td>3.84</td>
<td>25%</td>
</tr>
<tr>
<td>16-Bit RCA</td>
<td>30.72</td>
<td>5.76</td>
<td>18.75%</td>
</tr>
</tbody>
</table>

As seen in Table 3.8, the detectable HT size ratio fluctuates in range of 18.75-25%. To predict a practical detectable HT size with known testing-path size, equation (3.2)(3.3)(3.4) are used to achieve Linear Regression function of detectable by-path HT/testing-path size ratio, as

\[
T = [P \quad 1] \times x^* = 0.1696P + 0.72
\]

(3.6)

where detectable by-path HT size is \( T \), detecting-path size is \( P \). Equation (3.6) and the three data points in Table 3.8 used in Linear Regression are plotted in Fig. 3.15. The straight-line slope is used to estimate the ratio of detectable by-path HT/testing-path size.
size, thereby we can claim that a by-path HT with size greater than 16.96% of HT-connected propagation path size can be detected by the proposed detection methodology.

3.3.5 Power-Overhead Analysis

The proposed detection technique will only be conducted off-line in conjunction with functionality/reliability testing with emphasis on detection of HT inserted during fabrication. The additional power consumed by the detection circuit in field would be determined with the detection circuitry in passive (sleeping) mode. Then for detection circuit power overhead measurement, the CUT is active while HT detection circuit is dormant. To find power-overhead of the proposed detection circuit, a power measurement experimentation is set up as follows. 4, 8, 16-bit RCAs are used as CUT, respectively. Power consumption of CUT is measured when CUT operates at frequency of 100MHz. Power consumption of CUT with detection circuit is measured when CUT operates at same frequency of 100MHz while detection circuit is dormant. Detection circuit in dormant mode will only consume static power. The static power consumption of the same single detection circuit is fixed regardless of the size of corresponding CUT. Therefore, power-overhead of detection circuit is inverse-proportional to the size of CUT. The power measurement results are shown in Table 3.9. As seen, the power consumption overhead of detection circuit is 1.2-3.7%, which is inverse-proportional to the corresponding CUT size. Power consumption of detection circuits in 4, 8, 16-bit RCAs are 0.8µW, 0.91µW, 0.9µW, respectively. All the three RCAs are paired with the same detection circuit. However, due to the different signals applied to detection circuit by CUT testing path (signal n1 and n3 in Fig. 3.4), the detection circuit is in different static power states thereby causing static power consumptions of detection circuit in CUTs are slightly different.
Table 3.9 Power Consumption of CUT & HT Detection Circuit

<table>
<thead>
<tr>
<th>CUT</th>
<th>Power Consumption (μW)</th>
<th>Power Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CUT</td>
<td>CUT &amp; Detection Circuit</td>
</tr>
<tr>
<td>4-Bit RCA</td>
<td>21.52</td>
<td>22.32</td>
</tr>
<tr>
<td>8-Bit RCA</td>
<td>39.85</td>
<td>40.76</td>
</tr>
<tr>
<td>16-Bit RCA</td>
<td>74.12</td>
<td>75.02</td>
</tr>
</tbody>
</table>

3.4 Xilinx Implementation

An 8-bit digital finite impulse response filter (FIR) with order of 12 is implemented in FPGA as CUT for HT detection experiment [50]. Digital FIR, shown in Fig. 3.16, is a filter whose impulse response is of finite duration, because it settles to zero in finite time. Each value of the output sequence is a weighted sum of the most recent input values:

\[ y[n] = \sum_{i=0}^{N} b_i x[n - i] \]  (3.7)

where \( x[n] \) is input, \( y[n] \) is output, \( N \) is filter order, \( b_i \) is filter coefficient [51].

Fig. 3.16 Digital FIR Filter of Order \( n \)

The critical path of FIR is used as the HT testing-path. The XC6VLX240T FPGA board, shown in Fig. 3.17, is deployed in this experimentation. FPGA internal signal writing and reading are conducted using ChipScope, whose organization diagram is shown in Fig. 3.18. A clock at 200MHz frequency is generated by Phase Locked Loop (PLL) on board, then it is converted to required filter clock by Digital Clock Manager.
Integrated Controller (ICON) communicates with the host PC and sends commands to other ChipScope modules via a control port. The Virtual Input/Output (VIO) core is customized to drive a pulse train of successive values for CUT input signal. CUT input, CUT output and HT-detection signals on FPGA are monitored and send for display by Integrated Logic Analyzer (ILA) core.

![Fig. 3.17 FPGA Board (XC6VLX240T)](image)

![Fig. 3.18 FPGA ChipScope Organization](image)

The implementation results of FPGA, shown in Fig. 3.19, match the detection
algorithm: the detection signal remains low for Trojan-free CUT at second clock rising edge, while it is pulled up to ‘1’ for Trojan-attacked CUT at second clock rising edge.

![Waveform Image](image-url)

(a)

![Waveform Image](image-url)

(b)

Fig. 3.19 FPGA Implementation Result of FIR Filter

Fabricated circuit operating variations include process, voltage, and temperature (PVT) variation. To account for effect PVT variation on HT detection parameters, the HT detection method is implemented on ten FPGA boards. The implementation results are listed in Table 3.10, in which FPGA boards are ordered by CUT delay time. To achieve 10% FP (false alarm) for Trojan-free circuit, CUT clock period is set up as double of the ninth CUT delay (4.9ns), so that FPGA 1-9 can be correctly asserted to be Trojan-free. The eighth detectable HT size is the minimum detectable Trojan size at 90% detection probability with 10% probability of FP. As seen in Table 3.10, an in-path Trojan with size 0.7% of host-circuit and 3.2% of testing-path is detectable at detection probability 90% with 10% false alarm. HT size ratio is less than schematic implementation, because the limited FPGA chips have less process variation than
CMOS process modelling. The system clock frequency generated by DCM on FPGA board is not continuous, so CUT delay and detectable HT size are not as accurate as Cadence schematic implementation which can generate continuous clock frequency.

Table 3.10 Detectable HT Size in FPGA Boards

<table>
<thead>
<tr>
<th>FPGA Board</th>
<th>CUT Delay (ns)</th>
<th>In-Path HT</th>
<th>By-Path HT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>HT/Host-Circuit Ratio</td>
<td>HT/Testing-Path Ratio</td>
</tr>
<tr>
<td>1</td>
<td>4.7</td>
<td>0.5%</td>
<td>2.4%</td>
</tr>
<tr>
<td>2</td>
<td>4.7</td>
<td>0.5%</td>
<td>2.4%</td>
</tr>
<tr>
<td>3</td>
<td>4.7</td>
<td>0.5%</td>
<td>2.4%</td>
</tr>
<tr>
<td>4</td>
<td>4.8</td>
<td>0.7%</td>
<td>3.2%</td>
</tr>
<tr>
<td>5</td>
<td>4.8</td>
<td>0.7%</td>
<td>3.2%</td>
</tr>
<tr>
<td>6</td>
<td>4.8</td>
<td>0.7%</td>
<td>3.2%</td>
</tr>
<tr>
<td>7</td>
<td>4.8</td>
<td>0.7%</td>
<td>3.2%</td>
</tr>
<tr>
<td>8</td>
<td>4.9</td>
<td>0.7%</td>
<td>3.2%</td>
</tr>
<tr>
<td>9</td>
<td>4.9</td>
<td>0.7%</td>
<td>3.2%</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>0.9%</td>
<td>4%</td>
</tr>
</tbody>
</table>

3.5 Experimental Evaluation

This section clarifies three terms in this chapter: testing-path, host-circuit and host-chip. They are important in understanding the HT detection parameters. Thereafter benchmark circuits are used to evaluate the effectiveness of the proposed detection method.
3.5.1 HT Detection Parameters

As mentioned before, the proposed HT detection circuit is applied on a selected testing-path, which is a part of CUT. The *host-circuit* in the aforementioned detection implementations is defined as the circuit associated with the testing-path. A low-power carry-select-adder (CSA), shown in Fig. 3.20, is used to illustrate the HT detection parameters for Trojan associate with testing-path, host-circuit and the whole host-chip [52]. In this experiment, a HT circuit is randomly embedded in the testing-path, which is from $C_{in}$ to $C_{out}$ in Fig. 3.20. With detection probability of 90% and 10% probability of FP, the minimum detectable Trojan size is $0.24 \mu m$, which is 14.3% of the testing-path size, 1.85% of the host-circuit size, and 0.09% of the host-chip (CSA) size. The testing-path size is the size summation of transistors in CUT that directly connected to the testing-path. The host-circuit size is the size of circuit in CUT that associate with the testing-path, which is the circuit in dotted box in Fig. 3.20. The host-chip size in this case is the size of CSA.

![Fig. 3.20 Low-Power Carry-Select-Adder](image)

The detectable Trojan size ratio in section 3.3.3 and 3.3.4 is the ratio of Trojan size over testing-path size or host-circuit size. Based on the implementations in section 3.3.3 and 3.3.4, the detectable Trojan size is one or two orders smaller than testing-path...
or host-circuit size. As discussed above, the ratio of detectable Trojan size over the whole host-chip size is even much less. Also, the detectable Trojan size and detection probability can be further improved by deploying more detection circuits on the testing-path. For example, one detection circuit applied on the testing-path in Fig. 3.20 can detect Trojan with size of 0.24\(\mu\)m at detection probability 90%. Two detection circuit sequentially applied on the same testing-path will either detect Trojan with size of 0.12\(\mu\)m at detection probability 90% or detect Trojan with size of 0.24\(\mu\)m at detection probability 100%.

As mentioned, the selection of testing path paired with one detection circuit is determined by chip designer and testing engineer based on HT resistance to detection, the required HT detection probability, the probability of false alarm, and limitations on parameter overhead.

### 3.5.2 Experimental Result

ISCAS-85/89 benchmarks are used to evaluate the effectiveness of the proposed detection method. Critical paths in benchmarks are as testing-paths. Inverter buffers are randomly embedded in testing-path to account for HT. Multiple detection circuits are applied on each testing-path to achieve that the minimum detectable HT size is about 2% of host-circuit size. Operating clock frequency and HT size are set when FN and FP are equal to 5%. The simulation result is shown in Table 3.11. As seen, the minimum detectable HT size is 1.32-3.85% of host-circuit size that is dependent on host-circuit design.

<table>
<thead>
<tr>
<th>CUT</th>
<th>Gate</th>
<th>(f_{\text{clock}}) (GHz)</th>
<th>HT Size ((\mu)m)</th>
<th>HT/Host-Circuit Ratio</th>
<th>Area Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>S27</td>
<td>13</td>
<td>1.45</td>
<td>0.24</td>
<td>3.85%</td>
<td>3.94%</td>
</tr>
<tr>
<td>S344</td>
<td>175</td>
<td>0.31</td>
<td>1.68</td>
<td>2.39%</td>
<td>2.11%</td>
</tr>
</tbody>
</table>
Table 3.12 lists parameters of some recently published timing analysis-based HT detection results. With similar detection probability, the proposed technique is competitive in HT detection accuracy and sensitivity with relatively low overhead. Note that the detectable HT size can be decreased by adding more detection circuits in one testing-path, but its attendant more overhead.

Table 3.12 Comparison of HT Detection Methods

<table>
<thead>
<tr>
<th>Reference</th>
<th>Detection Probability</th>
<th>False Positive</th>
<th>Detectable HT Size</th>
<th>Area Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>This Work</td>
<td>95%</td>
<td>5%</td>
<td>0.06-0.22%&lt;sup&gt;a&lt;/sup&gt;</td>
<td>1.3-3.94%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.32-3.85%&lt;sup&gt;b&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>Ngo [39]</td>
<td>95%</td>
<td>5%</td>
<td>1.7%&lt;sup&gt;a&lt;/sup&gt;</td>
<td>0</td>
</tr>
<tr>
<td>Kelly [53]</td>
<td>100%</td>
<td>0</td>
<td>0.3%&lt;sup&gt;a&lt;/sup&gt;</td>
<td>7.5%</td>
</tr>
<tr>
<td>Nejat [41]</td>
<td>97%</td>
<td>16%</td>
<td>6.1%&lt;sup&gt;b&lt;/sup&gt;</td>
<td>20%</td>
</tr>
</tbody>
</table>

<sup>a</sup> minimum detectable HT size/host-chip size

<sup>b</sup> minimum detectable HT size/host-circuit size

3.6 Conclusion

A novel HT detection technique based on timing analysis is proposed. The detection process of each testing-path takes one clock cycle. The detection method can be adopted in both combinational and sequential circuit. Probability of detection FP (false alarm) is effectively determined by the test clock frequency. For 90nm CMOS ASIC benchmarks, the ratio of detectable HT size over host-circuit size ranged from 2.81% to 3.37% at 90% detection probability with 10% FP. For FPGA implementation
of HT detection, the ratio of detectable HT size to host-circuit size ranged from approximated 0.5% to 0.9% at 90% detection probability. The detectable HT size and detection probability can be further improved by applying more detection circuits on a testing-path. Also, ring-oscillator can be introduced to estimate operation temperature and process variation of testing-path to calibrate detection parameters, eventually improve detection probability.
4 Self-Reference-based Hardware Trojan Detection

A self-reference-based power-analysis microelectronic circuit Hardware Trojan (HT) detection methodology is proposed in this chapter. Simulation results show that the proposed technique can detect HTs with areas that are 0.013% of the host-circuitry. ISCAS benchmark circuits are used to evaluate efficiency of the developed method. The discussion in this chapter is substantially drawn from [54][55], where we first reported the development and evaluation of this technique.

4.1 Introduction

Recently, HT detection based on power analysis has arisen as an effective technique. Embedded HT will add current paths and loads to the original circuit, that result in extra power consumption on wires and gates in Trojan affected area. HT can be revealed by measuring the differential power characteristics of the attacked circuit.

Most publications based on power analysis techniques focus on algorithms to distinguish the power difference caused by HT and methods to augment this difference. In [27], authors distinguish the HT-attacked chip by comparing the power traces of circuit-under-test (CUT) with genuine circuit. Statistical methods are used to improve the detection accuracy. This method is practical on HT with sufficient size to cause detectable differences, however HT is normally designed with small size to maintain the stealthy nature. It is difficult to directly distinguish the tiny effect of HT on a modern complex chip. To augment the effect of HT, authors in [56] partition circuit into regions using scan chains, then specific test vectors are generated to magnify the activity in the target region. The power trace of the CUT is compared to golden parameters to ascertain security reliability. Results show that the impact of HT on host-circuit is effectively enlarged by this method. However, due to the instability of golden standards caused by
inter and intra-die variations and the maskable tiny power of HT caused by intrinsic process and environmental variations of the host-chip, this method becomes practical only for relatively large HT circuit designs. In [57], authors proposed a HT detection method based on gate level characterization (GLC), in which a genuine-guaranteed chip is not required. For each gate on chip, a manufacturing variability (MV) factor is multiplied with ideal static power of the gate to account for MV effect on power profile. To achieve more accurate results, thermal condition is considered for static power calculation. Thereafter, a controlled input vector switching is used to adjust the thermal conditioning of the entire chip to an expected level. A linear equation set is formulated by the sum of calculated static power of each gate added a single HT variable is equal to the measured static power of the whole chip. The HT variable is solved by a linear program solver. The CUT is affirmed as Trojan-free if HT variable is close to 0; otherwise, it is Trojan-attacked. To increase the detection accuracy in [57], CUT is partitioned into segments using input vector selection in [58]. When the static power of a segment is measured, a set of input vectors are applied to alter the static status of the target segment while other segments are dormant.

A self-reference-based Trojan detection method is proposed in this chapter, in which the effect of operating variations can be reduced by referencing neighbor elements. The experimental results show that the HT detection accuracy is competitive compared to state-of-the-art.

4.2 Preliminaries

This section introduces the basic design theory of the self-reference-based HT detection.
4.2.1 Self-Reference-based HT Detection Theory

A sequential circuit is used as an example to explain this proposed technique. As shown in Fig. 4.1, a CUT is partitioned into N segments excluding the system clock-tree. The clock-tree is an essential network in modern chips to distribute clock to all clocked elements at virtually the same time [59]. All partitioned segments in the CUT must be controllable by primary inputs. Each segment uses a separate power rail, and an on-chip clock-tree is powered by a specific rail in segment 0. To measure one segment static power, the voltage supply is applied to this segment along with the clock-tree rail, while leaving the other power rails float.

![Segment Diagram]

Fig. 4.1 Partitioned CUT

Considering an integrated circuit (IC) segment $q$ with input pattern $IN$, the static power measurement can be modeled as consisting of following components: (1) ideal static power of component $i$ in $q$, $p_i(q, IN)$ which is dependent on operating segment and input pattern. The ideal power is obtained by simulation based on tt (typical-typical) corner analysis; (2) operating variations (process, voltage, temperature) effect on static power of component $i$, $n_{vt}(q, IN, t)$ which is a random value, but it is influenced by

70
operating segment, input pattern, and operating time \((t)\); (3) signal noise \(n_s(q, IN, t)\) which depends on operating segment, input pattern, and operating time; (4) extra power consumed by Trojan circuit \(p_T(q, IN, t)\) if segment is attacked by Trojan circuit. Therefore, static power measurement of Trojan-free segment \(q\) with \(k\) components can be modeled by (4.1)

\[
P_{mes}(q, IN, t) = \sum_{i=1}^{k} (p_i(q, IN) + n_{vi}(q, IN, t)) + n_s(q, IN, t)
\]

and static power of the Trojan-attacked \(q\) is

\[
P_{mes}(q, IN, t) = \sum_{i=1}^{k} (p_i(q, IN) + n_{vi}(q, IN, t)) + n_s(q, IN, t) + p_T(q, IN, t)
\]

The random signal noise \(n_s(q, IN, t)\) can be largely eliminated by averaging over a number of power measurements for the same IC. So signal noise is ignored in (4.1) and (4.2). Fabricated circuit operating variations include process, voltage, and temperature (PVT) variation. The same component in \(q\) will have the same process and voltage variation for different input pattern \(IN\). Different \(IN\) will result in different power consumption and eventually vary operating temperature. A specific operating variable scaling factors \((\lambda)\) is introduced to account for the average PVT variation on a circuit component compared to the ideal static power consumption. The static power of genuine \(q\) and Trojan-attacked \(q\) are modeled as (4.3) and (4.4), respectively.

\[
P_{mes}(q, IN, t) = \sum_{i=1}^{k} \lambda_i p_i(q, IN)
\]

\[
P_{mes}(q, IN, t) = \sum_{i=1}^{k} \lambda_i p_i(q, IN) + p_T(q, IN, t)
\]

Equation (4.4) accounts for the extra Trojan power consumption, but is not used in the self-reference-based detection methodology. Equation (4.3) is used as power measurement model in the rest of this chapter. Inserting Trojan will increase power consumption \((P_{mes})\) in (4.3), meanwhile \(p_i\) is fixed from tt corner simulation. Therefore, the \(\lambda\) value in Trojan-attached circuit will be increased compared to Trojan-free circuit.
The proposed HT detection methodology is based on solving for each component \( \lambda \) and clock-tree \( \lambda \) with the pre-determined (simulated) ideal static power \( (p_i) \) of each component and measured power \( (P_{mes}) \) of each segment with different input patterns. HT-attacked assertion in each segment is based on the \( \lambda_{clk} \) for that segment falling outside of prescribed limits. The prescribed limit of \( \lambda_{clk} \) for a specific segment is dependent on reference of the values of \( \lambda_{clk} \) in other segments of the chip. So, this technique is denoted as self-reference-based HT detection.

Basically, this detection method uses a global component to estimate the parametric stability of each partitioned segment. The clock-tree is used as a global component in this chapter, since most practical chip designs including both combinational and sequential circuits are with a clock-tree to meet modern circuit requirements, high operating frequency and high resolution. For detection on combinational circuit only chips without clock-tree, this method needs to use other global circuits as reference component, such as a power-gating control circuit.

The main contributions of this technique are as follows:

- HT detection method can be applied with zero-overhead.
- Genuine chip is not needed.
- Location of HT can be estimated.
- Partitioned CUT increases detection accuracy.
- Detection accuracy is adjustable by size of partitioned segments.
- Computational complexity is optimizable based on required detection accuracy.

There is a potential I/O issue with this technique because each partitioned segment requires a separate pair of power rail \( (V_{DD}, GND) \), which occupies two I/O pins. So, the number of required power pins may increase with the number of partitioned segments. However, to have a stable power source, chip designers usually provide a
number of I/O pads for power rails. For example, in MIPS processor, 11 out of 40 I/O pads are used for power rails [60][61]. Then the existing power pads can be employed to separate power segments. Furthermore, a multiplexer can be exploited to control more power rails for segments with limited I/O pins. To achieve the maximum detection accuracy with zero overhead, all existing power pads should be exploited to separate power segments, that means the number of partitioned segments ($N$) should follow $2 * N = n_p$ (or $2 * N = n_p - 1$ if $n_p$ is odd), where $n_p$ is the number of power pads. Note that chip designer can further increase detection accuracy by partition circuit into more segments, but its attendant overhead by extra control circuit splitting the existing power rails.

Compared to the conventional practical HT detection methodologies using variation factors and portioned CUT, the uniqueness of the proposed method is stated as follows:

- The power measurement of each partitioned segment is more accurate, that will benefit the HT detection probability. In the proposed method, when the power of a circuit segment is measured, other circuit segments are physically isolated from power rails. The conventional methods either emphasize the power of target segment by magnifying the activity of target region or achieve the power of partitioned segment by mathematic analysis [56][58].

- The operating variable scaling factors ($\lambda$) of a global component is used to estimate the parametric stability of each partitioned segment. The segment is asserted as Trojan-attached if $\lambda$ of global component exceeds prescribed limit in it. It is more efficient compared to conventional HT detection method using variation factors, which requires massive math analysis for the scaling factor of each gate on chip.
• HT detection overhead and workload can be adjusted by chip designer based on vulnerability of path, the desired HT resistance, and limitations on parameter overhead. For example, the security-sensitive portion of circuit (e.g., memory) is required to be covered by segment with smaller size (e.g., number of gates) and more power profiles need to be measured to achieve more accurate HT detection. The reason why more power profiles benefit detection accuracy is explained by Fig. 4.6 in section 4.3. Memory circuit is security-sensitive, because the key data stored in memory could be a target for attackers. While security-robust portion of circuit (e.g., critical path) can be covered by larger segment and less power measurements can be conducted to reduce workload and circuit overhead. Critical path is security-robust, because any extra connection on critical path will slow chip operating frequency, which can be easily revealed in post-fab functional test. Moreover, if detection accuracy is not critical for a segment, the operating variation scaling factor ($\lambda$) can be used for a group of multiple gates instead of single gate to tremendously reduce the computational complexity of the detection methodology, which is introduced in section 4.3.3.

4.2.2 Operating Variation

Because of variations of physical parameters and fabrication process parameters, transistors may have different threshold voltage ($V_t$) and channel length ($L$). In addition, interconnect lines may have different width, thickness, and contact resistance. All these can affect circuit design timing and power parameters. This situation is denoted as process variation that can cause up to 30% circuit performance variation in 90nm technology [62].

Power supply voltage may vary up to 10% due to capacity of voltage regulator, IR drops along supply rail, and noise. This will further alter the power consumption and
delay of a designed circuit [46].

Also, the increased logic density and operating frequency of modern ICs have exponentially magnified operational variations due to temperature fluctuations. Increased temperature reduces transistor switching speed meanwhile increases transistor leakage current, that increases chip power consumption and propagation delay.

Normally, HTs are designed to remain dormant until activation to avoid detection. However, the embedded HT causes leakage current even in sleeping mode that contributes to static power consumption of host-circuit. Therefore, detection of an HT in the sleeping mode is dependent on static power consumption. Static power is negligible compared to dynamic power, so circuit activity causing dynamic power will dramatically reduce the power contribution of dormant HT on host-circuit. Also, PVT variations affect static power to a lesser extent than dynamic power. Circuit static power follows a Gaussian distribution as a function of PVT variations with the magnitude of the 3σ standard deviation about 19% of the mean value [63]. Retaining parts within a 3σ distribution will result in 0.26% of parts being rejected.

To check whether the input patterns and PVT variations affect circuit static power or not, a power measurement experimentation is set up as follows. A 1-bit full adder (FA) is simulated with all eight input patterns using Monte Carlo analysis to account for PVT variations. Three input patterns are randomly picked from Monte Carlo analysis (50 iterations for each input pattern). For each of the 50 Monte Carlo simulations, a ratio is calculated which is equal to the Monte Carlo simulated power consumption (measured power) over the designed power consumption (ideal simulated result with typical/typical model parameters). This ratio is calculated for static power consumption of an AND gate (G1) in the FA and is plotted in Fig. 4.2. Each axis represents measured over designed static power of G1 for each input pattern. Each
bubble is located by measured power over ideal power of $G1$ in the same Monte Carlo iteration with each of the three possible input pattern simulations. All dots on the red solid line in Fig. 4.2 have equal x, y, z values (same ratio of measured to simulated power for all three input patterns). As seen in Fig. 4.2, all bubbles distribute along the red line, but vary from 0.8 to 1.4 as PVT variation, which indicates the power deviation changes greatly with Monte Carlo PVT variations, but remains relatively close to the same value for all three input patterns for the same Monte Carlo iteration (same PVT parameters).

![Deviation of Static Power (simulated/designed)](image)

**Fig. 4.2 Static Power Deviation of Gate $G1$**

As mentioned in section 4.2.1, to account for the PVT effect on static power consumption of any designed circuits, a scaling factor $\lambda_i$ is introduced to estimate the real static power of gate $i$ based on ideal simulated power as $\lambda_i p_{ij}$, where $p_{ij}$ is the ideal static power of gate $i$ with input vector $j$. Similarly, a scaling factor $\lambda_{clk}$ is used
to estimate the real static power of a clock-tree as $\lambda_{clk}p_{clk}$ where $p_{clk}$ is the ideal static power of the clock-tree.

4.3 Self-Reference-based HT Detection

This section introduces the proposed HT detection algorithm and mechanism. Benchmark circuits are used as examples to explain the detection methodology and optimization of computational complexity.

4.3.1 Self-Reference-based HT Detection Algorithm

The flowchart of self-reference-based HT detection algorithm consists of six steps as shown in Fig. 4.3.

![Flowchart](image)

Fig. 4.3 Overall Flow of Self-Reference-based HT Detection
1) Partition CUT into \( N \) segments, and ensure clock-tree is in a separate single segment. \( N \) circuit segments use \( N \) separated power rails. HT detection accuracy is proportional to the ratio of HT size to the segment size where HT is located. In other words, detection accuracy rises with number of partitioned segments (\( N \)) in CUT while all the partitioned segments are roughly equal in size. (Note size is estimated by total CMOS transistor width.)

2) Table the ideal (simulation data in tt corner) designed static power consumption of each gate with specified input patterns for each segment.

3) Power on one segment together with clock-tree, while power rails in other segments float. Measure static power consumption of CUT (\( P_{mes} \)) with different input patterns.

4) To determine the mean effect of PVT variations on designed circuits, the ideal static power consumption of clock-tree (\( p_{clk} \)) and gates (\( p_i \)) is multiplied by an operating variation scaling factor (\( \lambda \)). A set of \( m \) equations is formulated as shown in (4.5) with assumption that the testing segment consists of \( k \) gates with \( m \) static states. Each static state is a state with specific static power consumption.

\[
P_{catj} = \sum_{i=1}^{k} \lambda_i p_{ij} + \lambda_{clk} p_{clk}, \quad j = 1 \ldots m
\]

where \( P_{catj} \) is the static power consumption of CUT with static state \( j \), \( \lambda_i \) is scaling factor of gate \( i \), \( p_{ij} \) is ideal static power of gate \( i \) with static state \( j \). The unknown variables in (4.5) are \( \lambda_i \) and \( \lambda_{clk} \).

A combinational circuit with \( K_{in} \) primary inputs has \( 2^{K_{in}} \) possible number of static states. In a sequential circuit, registers are used to divide sections of combinational logic, as shown in Fig. 4.4. For the sequential circuit in Fig. 4.4 with \( K_{in} \) inputs, there are \( 2^{K_{in}} \) static states for each combinational logic. Primary input signals propagate through one section of combinational logic in each clock cycle. Each
section of combinational logic can operate on a specific primary input pattern controlled by clock. $M$ sections of combination logic can generate $2^{K_{in}M}$ static states. Also, clock may be ‘0’ or ‘1’ for a given input. Therefore, the number of static states for a sequential circuit with $K_{in}$ primary inputs and $M$ sections of combinational logic is $2 * 2^{K_{in}M}$.

Fig. 4.4 Sequential Circuit Structure

With equation set (4.5), the value of $\lambda_i$ and $\lambda_{clk}$ are estimated by finding the solution that minimizes the sum squared deviation of measured (Monte Carlo simulated static power) and calculated static power as shown in (4.6). The power variations of gates are constrained to +/- 30% of the ideal calculated values because PVT variation affects circuit static power consumption within 30% [63]. This means ($\lambda_i$) should be constrained to $0.7 < \lambda_i < 1.3$. Then finding the best estimate of $\lambda_{clk}$ becomes a linear optimization problem as follows

$$\text{minimize} \sum_{j=1}^{m} \left( P_{mesj} - P_{calj} \right)^2 = \sum_{j=1}^{m} \left( P_{mesj} - \left( \sum_{i=1}^{k} \lambda_i p_{ij} + \lambda_{clk} p_{clk} \right) \right)^2$$

subject to $0.7 < \lambda_i < 1.3$, $i = 1, ..., k$ (4.6)

To solve for the unknown variables in (4.6), the number of static states ($m$) can vary from 1 to $N_s$, where $N_s$ is the number of total feasible static states. Using more static states in (4.6) are expected to generate more accurate $\lambda$ by averaging the $\lambda$ fluctuation over more operating states. However, more static states require more power measurements and the calculation in (4.6) is more complicated. A simple benchmark circuit (CUT1) is used as an example to illustrate this process. As shown in Fig. 4.5, CUT1 consists pipelined 3-inputs AND, OR, XOR gate and is partitioned into three
segments plus one clock-tree segment. Each partitioned segment consists of two combinational gates and one sequential gate (a D flip-flop). To account for process variation, the benchmark circuit is simulated with Monte Carlo analysis including process variation and mismatch. The simulated Monte Carlo results are referred to as the measured values. Scaling factor of the clock-tree ($\lambda_{clk}$) is calculated with number of static states varying from 4 to 10. For each number of static states, there are $C_{N_s}^m$ static state combinations that can be chosen, where $N_s$ is the number of total feasible static states, and $m$ is number of selected static states. For example, all segments in CUT1 have 32 static states. For each segment, a set of 4, 5, ..., 10 static states are selected from the possible 32 states and $\lambda_{clk}$ deviation from ideal value of 1 resulting from solving (4.6) for each segment is plotted in Fig. 4.6.

Fig. 4.5 Partitioned CUT1
In Fig. 4.6, "\(\lambda_{clk}\) deviation" represents deviation of calculated \(\lambda_{clk}\) from ideal value of 1; \(\lambda_{clkq}\) is the \(\lambda_{clk}\) based on calculation in segment \(q\). As seen, a larger number of static states results in a solution that converges to a more accurate estimate of \(\lambda_{clk}\).

5) Iterate step 3, 4 for the remaining segments.

6) Define that deviation of clock-tree scaling factor in segment \(q\) is

\[
\Lambda_q = \frac{\lambda_{clkq} - \min(\lambda_{clk1}, \ldots, \lambda_{clkN})}{\min(\lambda_{clk1}, \ldots, \lambda_{clkN})}
\]

Constraints must be placed on the threshold of values for \(\Lambda_q\) for declaring a Trojan-free or Trojan-attached segment. Seven Trojan-free benchmark circuits are designed in IBM 90nm CMOS technology and simulated with Monte Carlo analysis (100 samples). The simulation results are plotted in Fig. 4.8, where false negative is the fault that assert Trojan-attacked for a Trojan-free circuit; CUT1, CUT2 are customized benchmarks, shown in Fig. 4.5 and Fig. 4.9, respectively; s27 and s344 are ISCAS-89 benchmarks; c499, c6288 and c5315 are ISCAS-85 benchmarks, shown in Fig. 4.7.
ISCAS-89 s344 is a 4-bit multiplier which consists of 16 *multiplexers*, 12 *D flip-flops*, 3 *T flip-flops*, one 4-bit *full adder*, and one 4-bit *AND* gate. ISCAS-85 c499 is a single-error-correcting circuit [64]. The 41 inputs are combined in module *M1* to form an 8-bit internal bus *S*, which then combines with 32 primary inputs in module *M2* consisting correcting logic. Benchmark c6288 is a 16*16* multiplier formed by a 15*16 matrix with 240 full and half adders. Benchmark c5315 is a 9-bit arithmetic logic unit (ALU) that performs arithmetic and logic operations simultaneously on two 9-bit input signals.
Fig. 4.7 ISCAS-89 Benchmarks (a) S27, (b) S344, and ISCAS-85 Benchmarks (c) c499, (d) c6288, (e) c5315
Fig. 4.8 Threshold of Clock-Tree Scaling Factor Deviation in Benchmarks

It can be seen in Fig. 4.8 that the HT detection false negative rate is inversely proportional to the value of threshold of $A_q$. A smaller value for the threshold of $A_q$ may have higher detection accuracy, while attendant a higher number of false negatives. Detection accuracy of Trojan-attacked circuit is highly dependent on the size-relationship of Trojan and host-circuit. Also, it is noted that false positive (assert Trojan-free for Trojan-attacked circuit) is related to both the host-circuit design and the unknown HT design. As seen in Fig. 4.8, benchmark circuits with threshold values of $A_q$ of 10% or more have no false negatives. The seven benchmarks do not represent all possible circuits; however, the selected benchmarks cover a range of a few to thousands of gates with a few to hundreds of IO pins form a basis for selecting a threshold for the value $A_q$. Based on these results, 10% is chosen as a minimum value for $A_q$ to declare a Trojan attacked segment. Thus, segment $q$ is identified as Trojan-attacked if $A_q \geq 10\%$, otherwise segment $q$ is Trojan-free. Also, segment $q$ is defined
as Trojan-attacked if $\lambda_{clk_q} > 1.3$ regardless of value obtained by (4.7), since the static power of a segment can vary by 30% due to unknown circuit operational variables. Moreover, it is asserted that the clock-tree is attacked if $\lambda_{clk}$ in all segments exceeds 1.19. This is based on discussion in section 4.2.2, where the static power $3\sigma$ variation scale is about 19%, and only 0.26% of genuine circuits will be rejected outside of $3\sigma$ range. This means the false positive is about 0.26% for HT detection when the HT is embedded in clock-tree.

The criteria for asserting a segment is Trojan attacked is summarized below:

$\Lambda_q \geq 10\%$ implies segment $q$ is attacked;

$\lambda_{clk_q} > 1.3$ implies segment $q$ is attacked;

$\lambda_{clk} > 1.19$ in all segments implies clock tree is attacked.

The HT detection algorithm is abstracted as follows.
HT detection algorithm

**Input:** CUT, ideal static power of each gate in CUT \((p)\)

**Output:** presence and location of HT

1. Partition CUT into \(N\) segments;
2. for each segment \((q)\) do
3.    Power on the segment, while power rails in other
        segments are floated;
4.    for designed input patterns \((j)\) do
5.        Measure static power of CUT \((P_{mes_j})\)
6.    Find \(\lambda_{clkq}\) by minimizing
4\[ \sum_{j=1}^{m} \left( P_{mes_j} - \left( \sum_{i=1}^{k} \lambda_i P_{ij} + \lambda_{clk} P_{clk} \right) \right)^2 \]
   subject to \(0.7 < \lambda_i < 1.3, i = 1, ..., k\)
7. if \(\lambda_{clk} > 1.19\) in all segments then
8.    HT is present in clock tree;
9. for each segment \((q)\) do
10. \(\Lambda_q = \frac{\lambda_{clkq} - \min(\lambda_{clk1}, ..., \lambda_{clkN})}{\min(\lambda_{clk1}, ..., \lambda_{clkN})}\),
11. if \(\Lambda_q \geq 10\%\) then
12.    HT is present in segment \(q\);
13. if \(\lambda_{clkq} > 1.3\) then
14.    HT is present in segment \(q\);
15. **Return** HT presence and location;
The proposed technique can be used for post-fab HT detection or making watermark for individual chip. The HT detection benefits semiconductor manufacturing by revealing malicious alteration of chip and evading potential harms to client. As mentioned in section 4.2, because of inter-chip variations, the operating parameters of a single gate in different chips may be varied. Therefore, the operating variable scaling factors ($\lambda$) of each gate in a chip can form up a unique identification (ID) for that chip which is also denoted as watermark. Chip watermark can be used to protect intellectual property (IP) against illegal counterfeiting.

4.3.2 Example

The proposed HT detection method is illustrated in detail with a custom designed benchmark circuit (CUT1). This CUT consists of pipelined 3-inputs AND, OR, XOR gate, and partitioned as shown in Fig. 4.5. As aforementioned, the clock-tree is partitioned into a single segment, and separated power rails are applied to each segment.

As discussed in section 4.3.1, each segment in CUT1 may have 32 static states. To simplify the related calculation, 8 of them are picked with signal ‘clk’ equal to ‘1’. With the selected 8 static states, ideal static power consumption of each gate is shown in Table 4.1.

<table>
<thead>
<tr>
<th>Segment 1</th>
<th>Input Pattern</th>
<th>Gate Power (nW)</th>
<th>$P_{mes}$ (nW)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>Segement 1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
The circuits in segment 1 are powered on together with clock-tree in segment 0, while floating the power rails in other segments. Then the actual total static power consumption of CUT1 with the specific 8 static states ($P_{mes}$) is measured (simulate with Monte Carlo analysis), that is listed in Table 4.1. There are three columns below $P_{mes}$ in Table 4.1, “TF”, “T1”, “T0”, that represent total power consumption of CUT1 with
Trojan-free, Trojan embedded in segment 1, and Trojan-embedded in segment 0, respectively. An inverter which bypasses original circuit path is used as the Trojan.

For the Trojan free case, equation set (4.8), (4.9), (4.10) are created for segment 1, 2, 3, respectively, based on the calculated total static power of CUT1 (\(P_{\text{cal}}\)) is equal to sum of ideal static power of components for each static state.

\[
\begin{align*}
P_{\text{cal}1} &= 0.27\lambda_1 + 0.27\lambda_2 + 0.27\lambda_3 + 0.93\lambda_{\text{clk}1} \\
P_{\text{cal}2} &= 0.27\lambda_1 + 0.27\lambda_2 + 0.27\lambda_3 + 0.93\lambda_{\text{clk}1} \\
P_{\text{cal}3} &= 0.36\lambda_1 + 0.27\lambda_2 + 0.27\lambda_3 + 0.93\lambda_{\text{clk}1} \\
P_{\text{cal}4} &= 0.51\lambda_1 + 0.35\lambda_2 + 0.27\lambda_3 + 0.93\lambda_{\text{clk}1} \\
P_{\text{cal}5} &= 0.27\lambda_1 + 0.27\lambda_2 + 0.36\lambda_3 + 0.93\lambda_{\text{clk}1} \\
P_{\text{cal}6} &= 0.27\lambda_1 + 0.27\lambda_2 + 0.36\lambda_3 + 0.93\lambda_{\text{clk}1} \\
P_{\text{cal}7} &= 0.36\lambda_1 + 0.27\lambda_2 + 0.36\lambda_3 + 0.93\lambda_{\text{clk}1} \\
P_{\text{cal}8} &= 0.51\lambda_1 + 0.35\lambda_2 + 0.51\lambda_3 + 0.93\lambda_{\text{clk}1}
\end{align*}
\]

(4.8)

\[
\begin{align*}
P_{\text{cal}1} &= 0.75\lambda_1 + 0.27\lambda_2 + 0.75\lambda_3 + 0.93\lambda_{\text{clk}2} \\
P_{\text{cal}2} &= 0.49\lambda_1 + 0.35\lambda_2 + 0.49\lambda_3 + 0.93\lambda_{\text{clk}2} \\
P_{\text{cal}3} &= 0.82\lambda_1 + 0.35\lambda_2 + 0.49\lambda_3 + 0.93\lambda_{\text{clk}2} \\
P_{\text{cal}4} &= 0.49\lambda_1 + 0.35\lambda_2 + 0.49\lambda_3 + 0.93\lambda_{\text{clk}2} \\
P_{\text{cal}5} &= 0.75\lambda_1 + 0.27\lambda_2 + 0.82\lambda_3 + 0.93\lambda_{\text{clk}2} \\
P_{\text{cal}6} &= 0.49\lambda_1 + 0.35\lambda_2 + 0.49\lambda_3 + 0.93\lambda_{\text{clk}2} \\
P_{\text{cal}7} &= 0.82\lambda_1 + 0.35\lambda_2 + 0.49\lambda_3 + 0.93\lambda_{\text{clk}2} \\
P_{\text{cal}8} &= 0.49\lambda_1 + 0.35\lambda_2 + 0.49\lambda_3 + 0.93\lambda_{\text{clk}2}
\end{align*}
\]

(4.9)

\[
\begin{align*}
P_{\text{cal}1} &= 0.65\lambda_1 + 0.27\lambda_2 + 0.65\lambda_3 + 0.93\lambda_{\text{clk}3} \\
P_{\text{cal}2} &= 0.37\lambda_1 + 0.35\lambda_2 + 0.37\lambda_3 + 0.93\lambda_{\text{clk}3} \\
P_{\text{cal}3} &= 0.79\lambda_1 + 0.35\lambda_2 + 0.37\lambda_3 + 0.93\lambda_{\text{clk}3} \\
P_{\text{cal}4} &= 0.5\lambda_1 + 0.27\lambda_2 + 0.65\lambda_3 + 0.93\lambda_{\text{clk}3} \\
P_{\text{cal}5} &= 0.65\lambda_1 + 0.27\lambda_2 + 0.79\lambda_3 + 0.93\lambda_{\text{clk}3} \\
P_{\text{cal}6} &= 0.37\lambda_1 + 0.35\lambda_2 + 0.5\lambda_3 + 0.93\lambda_{\text{clk}3} \\
P_{\text{cal}7} &= 0.79\lambda_1 + 0.35\lambda_2 + 0.5\lambda_3 + 0.93\lambda_{\text{clk}3} \\
P_{\text{cal}8} &= 0.5\lambda_1 + 0.27\lambda_2 + 0.79\lambda_3 + 0.93\lambda_{\text{clk}3}
\end{align*}
\]

(4.10)

To find the best estimate of the scaling factor of clock-tree (\(\lambda_{\text{clk}1}\)), \(\sum_{j=1}^{8}(P_{\text{cal},j} - P_{\text{sim},j})^2\) is minimized subject to \(0.7 < \lambda_1, \lambda_2, \lambda_3 < 1.3\). Excel solver is used to solve the linear optimization problem. Then steps are repeated for the other two segments to obtain \(\lambda_{\text{clk}}\). The resulting estimates for \(\lambda\) for each segment are listed in
Table 4.2. As seen, the minimum $\lambda_{clk}$ is 0.87 in segment 1. The deviation of clock-tree scaling factor in segment 2 and 3 are calculated using (4.7) and meet the Trojan-free criteria as follows.

$$A_2 = \frac{\lambda_{clk2} - \lambda_{clk1}}{\lambda_{clk1}} = \frac{0.95 - 0.87}{0.87} = 9.2\% < 10\%$$

$$A_3 = \frac{\lambda_{clk3} - \lambda_{clk1}}{\lambda_{clk1}} = \frac{0.92 - 0.87}{0.87} = 5.7\% < 10\%$$

So, CUT1 is asserted as Trojan-free.

Table 4.2 Variation Scaling Factors in Trojan-Free CUT1

<table>
<thead>
<tr>
<th></th>
<th>$\lambda_1$</th>
<th>$\lambda_2$</th>
<th>$\lambda_3$</th>
<th>$\lambda_{clk}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segment 1</td>
<td>1</td>
<td>1.3</td>
<td>1.01</td>
<td>0.87</td>
</tr>
<tr>
<td>Segment 2</td>
<td>0.97</td>
<td>1.16</td>
<td>1</td>
<td>0.95</td>
</tr>
<tr>
<td>Segment 3</td>
<td>0.94</td>
<td>1.3</td>
<td>0.99</td>
<td>0.92</td>
</tr>
</tbody>
</table>

To evaluate the effectiveness of the proposed method in detecting Trojan-attacked circuit, this technique is applied while an inverter is embedded in segment 1 to account for HT. Simulated power consumption of CUT1 is listed in Table 4.1, “T1” column. As seen in Table 4.3, the minimum $\lambda_{clk}$ is 0.92 in segment 3. Deviation of clock-tree scaling factor in segment 1 and 2 are as follows,

$$A_1 = \frac{\lambda_{clk1} - \lambda_{clk3}}{\lambda_{clk3}} = \frac{1.42 - 0.92}{0.92} = 54.3\% > 10\%$$

$$A_2 = \frac{\lambda_{clk2} - \lambda_{clk3}}{\lambda_{clk3}} = \frac{0.95 - 0.92}{0.92} = 3.3\% < 10\%$$

So, CUT1 is defined as Trojan-attacked, and the Trojan is likely to be located in segment 1.

Table 4.3 Variation Scaling Factors of Clock-Tree

<table>
<thead>
<tr>
<th></th>
<th>$\lambda_{clk1}$</th>
<th>$\lambda_{clk2}$</th>
<th>$\lambda_{clk3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trojan-Free CUT</td>
<td>0.87</td>
<td>0.95</td>
<td>0.92</td>
</tr>
<tr>
<td>Trojan Embedded in Segment 1</td>
<td>1.42</td>
<td>0.95</td>
<td>0.92</td>
</tr>
</tbody>
</table>
To account for multiple HTs existing in multiple segments or single HT splitting in multiple segments, this technique is applied while an inverter is embedded in both segment 1 and segment 2 to imitate HT. As seen in Table 4.3, the minimum $\lambda_{clk}$ is 0.92 in segment 3. Deviation of clock-tree scaling factor in segment 1 and 2 are as follows,

$$
\Lambda_1 = \frac{\lambda_{clk1} - \lambda_{clk3}}{\lambda_{clk3}} = \frac{1.42 - 0.92}{0.92} = 54.3\% > 10\%
$$

$$
\Lambda_2 = \frac{\lambda_{clk2} - \lambda_{clk3}}{\lambda_{clk3}} = \frac{1.45 - 0.92}{0.92} = 57.6\% > 10\%
$$

So, CUT1 is defined as Trojan-attacked, and the Trojan is likely to be located in segment 1 and segment 2.

To evaluate HT detection for a clock-tree, this technique is applied while an inverter is embedded in segment 0 to account for HT. The calculated variation scaling factors of clock-tree ($\lambda_{clk}$) in Trojan-free and Trojan-attacked CUT are shown in Table 4.3. As seen, when Trojan is embedded in segment 0, $\lambda_{clk}$ in segments 1, 2, 3 all exceed 1.19, that point to an HT attacked clock-tree in CUT1.

### 4.3.3 Calculation Optimization

This section introduces a methodology for reducing the computational complexity in this detection technique. As seen in the proposed algorithm step 4, (4.6) consists $k + 1$ variables, $k$ is number of gates in the specific segment. Reducing variables can simplify the calculation of (4.6), however detection accuracy is sacrificed. In order to reduce the variables, the operating variation scaling factor ($\lambda$) can be used for a group of multiple gates rather than a single gate. Then the number of variables in a testing segment is reduced from number of gates to number of gate groups. A
benchmark circuit (CUT2) consists ISCAS-85 c17, 2-1 MUX, 1-bit FA, as shown in Fig. 4.9, is used to illustrate this situation.

Fig. 4.9 Partitioned CUT2

For segment 1 in CUT2, 7 gates result in 8 variables in each equation in (4.6), \(\lambda_1 - \lambda_7, \lambda_{clk1}\). However, if two gates are grouped to share one \(\lambda\), as shown in Fig. 4.9, the 4 gate groups generate 5 variables in (4.6), \(\lambda_1 - \lambda_4, \lambda_{clk1}\).

To verify how the size of gate group will affect detection accuracy, the proposed detection method is applied to CUT2 without/with HT embedded in segment 1, respectively, for gates grouped with size of 1, 2, 3 and 4. The resulting estimates for \(\lambda_{clk}\) in each segment are listed in Table 4.4, in which “Detectable HT Size” is the percentage of minimum detectable HT size over host-circuit size, and “Gate Group Size” is number of gates in each gate group. As seen, the increasing gate group size attendant increased detectable HT size that harms HT detection probability.
Table 4.4 Variation Scaling Factors in CUT2 with Different Gate Group Size

<table>
<thead>
<tr>
<th>Gate Group Size</th>
<th>$\lambda_{clk1}$</th>
<th>$\lambda_{clk2}$</th>
<th>$\lambda_{clk3}$</th>
<th>Detectable HT Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HT-Free</td>
<td>HT-Attacked</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1.006</td>
<td>1.338</td>
<td>1.017</td>
<td>1.028</td>
</tr>
<tr>
<td>2</td>
<td>0.981</td>
<td>1.421</td>
<td>0.983</td>
<td>0.997</td>
</tr>
<tr>
<td>3</td>
<td>0.954</td>
<td>1.756</td>
<td>0.965</td>
<td>0.997</td>
</tr>
<tr>
<td>4</td>
<td>0.998</td>
<td>1.36</td>
<td>1.077</td>
<td>0.997</td>
</tr>
</tbody>
</table>

Then the proposed detection method is applied to CUT2 with HT embedded in segment 0 (clock-tree segment) for gates grouped with different sizes. The results of HT in segment 0 and segment 1 are shown in Fig. 4.10, where “# variables” is number of variables in (4.6). It shows that with gate group size increasing, the number of unknown variables is decreased resulting in calculation simplification as indicated by circle markers (blue line). Also, the minimum detectable HT size is increased that sacrifices HT detection accuracy as shown by x and square markers (orange line). Detection for HT embedded in clock-tree is more difficult than for other segments of the chip, so the detectable size in a clock tree is typically larger than in logic segments. More on this in the next section.
4.3.4 HT Detection Overhead

The impacts of HT detection on area, power consumption, timing and signal noise of CUT are discussed in this section. The physical alteration of original circuit design by the proposed HT detection technique is power supply partitioning for individual segments. Typically, the power supply-related issues are power voltage drop and power supply noise, which are discussed as follows.

Power supply integrity is an essential part of chip design, because power-related issues can affect chip power consumption, timing and even lead to complete device failure. Systems are designed to operate at a nominal supply voltage, but the voltage may vary due to IR drops along supply rails [65]. This drop is \( \Delta V = I \times R \), where \( R \) is the supply rail resistance, \( I \) is the current along supply rail.
Chip power supply suffers from switching noise on the power supply rails, which is subsequently coupled onto the evaluation nodes of a circuit. To preserve signal integrity, every circuit design has a noise margin to allow limited signal variation. Extra noise on supply rails may exceed circuit noise margin, resulting in circuit functional failure. The power rail noise is proportional to its resistance \( R \).

Normally, practical chips are designed with a number of I/O pads for power rails to have a stable power source. When only the existing power pads are used to separate power segments \((2 \times N \leq n_p)\), none of extra circuit is required, and the power rails are divided for each single segment. Power rails are typically metal rings on higher-level routing layers, which carry power around the periphery of the die, around a standard cell’s core area, and around individual hard macros. Because the power rails are cut to shorter for individual circuit segment, the resistance \( R \) of each power rail is reduced. The current capability \( I \) of each partitioned power rail is less compared to the original power rail, because each partitioned power rail is only account for the corresponding sub-circuit segment. The reduced \( I \) and \( R \) can relieve power degradation \((\Delta V = I \times R)\) and noise (proportional to \( R \)), which will not deteriorate circuit operation parameters (e.g., power, timing, signal noise). And the circuit area is unaltered. Note that each partitioned power rail is required to be designed with capability to deliver sufficient current for the corresponding circuit segment.

Partitioning circuit into segments using more than existing power pads \((2 \times N > n_p)\) is not recommended, only if the detection accuracy is inadequate. Then extra control circuit is required to split the existing power rails for more segments. It is obviously increasing circuit area- and power-overhead, however because of the aforementioned reasons, timing and signal noise of CUT are not affected.
4.4 Experimental Result based on Monte Carlo Simulations

ISCAS-85/89 benchmarks are used to evaluate the effectiveness of the proposed detection method. Benchmarks are modified to sequential circuit by inserting registers in path. Inverter buffers are embedded as HT. Benchmarks are partitioned into four segments, in which segment 0 only consists clock-tree, and HT is randomly embedded in segment 1. The Trojan in each benchmark has a size of about 1% of host-circuitry. Gate-grouping is used to simplify detection calculation. Number of gates in each group varies from 4 to 35 depending on circuit size. The simulation result is shown in Table 4.5.

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Inputs/Outputs</th>
<th>Gate</th>
<th>$\lambda_{clk1}$</th>
<th>$\lambda_{clk2}$</th>
<th>$\lambda_{clk3}$</th>
<th>Trojan-Free</th>
<th>Trojan-Attacked</th>
<th>Detectable Trojan Size (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUT1</td>
<td>9/3</td>
<td>9</td>
<td>0.95</td>
<td>0.92</td>
<td>0.87</td>
<td>0</td>
<td>1.42</td>
<td>54.3%</td>
</tr>
<tr>
<td>CUT2</td>
<td>11/4</td>
<td>25</td>
<td>1.017</td>
<td>1.028</td>
<td>1.006</td>
<td>0</td>
<td>1.338</td>
<td>31.6%</td>
</tr>
<tr>
<td>S27</td>
<td>5/1</td>
<td>13</td>
<td>0.87</td>
<td>0.91</td>
<td>0.92</td>
<td>5.7%</td>
<td>1.33</td>
<td>52.9%</td>
</tr>
<tr>
<td>S344</td>
<td>9/11</td>
<td>175</td>
<td>0.94</td>
<td>0.97</td>
<td>0.96</td>
<td>2.1%</td>
<td>1.24</td>
<td>31.9%</td>
</tr>
<tr>
<td>C499</td>
<td>41/32</td>
<td>202</td>
<td>0.89</td>
<td>0.94</td>
<td>0.92</td>
<td>3.4%</td>
<td>1.31</td>
<td>47.2%</td>
</tr>
<tr>
<td>C6288</td>
<td>32/32</td>
<td>2406</td>
<td>1.014</td>
<td>1.007</td>
<td>1.001</td>
<td>0</td>
<td>1.21</td>
<td>20.2%</td>
</tr>
<tr>
<td>C5315</td>
<td>178/123</td>
<td>2406</td>
<td>1.013</td>
<td>0.987</td>
<td>1.01</td>
<td>2.3%</td>
<td>1.13</td>
<td>14.5%</td>
</tr>
</tbody>
</table>

As seen, deviation of clock-tree scaling factor in segment 1 ($A_1$) is within 10% in Trojan-free CUT, meanwhile $A_1$ is over 10% in Trojan-attacked segment 1. Therefore, the proposed detection technique is effective to distinguish HT in benchmark circuits. The minimum detectable Trojan size is 0.013-0.83% of host-circuit size, which relies
on two factors: 1) based on (4.7), the larger difference of $\lambda_{c_{lkq}}$ in the tested segment $q$ from minimum $\lambda_{c_{lk}}$ of the CUT results in smaller detectable Trojan size in segment $q$; 2) larger size of gate group causes larger detectable Trojan size. It is noted that the detectable Trojan size in the clock-tree falls near the maximum number of the detectable Trojan size range, while a Trojan in other segments falls close to the minimum number of the detectable Trojan size range. A HT embedded in the clock-tree is typically more difficult to detect than in other segments.

Table 4.6 lists parameters of some recently published power-analysis based HT detection results. The proposed self-reference method in this chapter is competitive in HT detection accuracy and sensitivity.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Modality</th>
<th>Detectable Trojan size</th>
<th>Area overhead</th>
<th>Trojan localization</th>
</tr>
</thead>
<tbody>
<tr>
<td>This Work</td>
<td>Static Power</td>
<td>0.013-0.83%</td>
<td>0</td>
<td>Yes</td>
</tr>
<tr>
<td>[66]</td>
<td>Power</td>
<td>3.9%</td>
<td>0</td>
<td>Yes</td>
</tr>
<tr>
<td>[1]</td>
<td>Transient Power</td>
<td>1-6%</td>
<td>0</td>
<td>Yes</td>
</tr>
<tr>
<td>[67]</td>
<td>Transient Current</td>
<td>1.6-2.3%</td>
<td>0</td>
<td>Yes</td>
</tr>
<tr>
<td>[68]</td>
<td>Transient Power</td>
<td>0.09-2.61%</td>
<td>0</td>
<td>Yes</td>
</tr>
<tr>
<td>[69]</td>
<td>Transient Power</td>
<td>0.1%</td>
<td>0</td>
<td>Yes</td>
</tr>
<tr>
<td>[70]</td>
<td>Static Power</td>
<td>0.4%</td>
<td>0</td>
<td>No</td>
</tr>
</tbody>
</table>

### 4.5 Conclusion

A self-reference-based microelectronic circuit Hardware Trojan detection methodology is proposed in this chapter. An embedded HT will contribute to circuit overall leakage/static power, which forms the basis for the self-referenced detection method, where a Trojan-free genuine circuit is not required. To account for the effect
of PVT variations on measured post-fabrication static power, the ideal model based static power consumption of the clock-tree ($p_{clk}$) and gates ($p_i$) is multiplied by an operating variation scaling factor ($\lambda$). The circuit is partitioned into segments, and a set of $m$ equations is formulated with assumption that the testing segment consists of $k$ gates with $m$ static states. The best estimate of $\lambda_{clk}$ is found by minimizing the sum squared difference of the calculated static power consumption and the measured static power consumption under constraints for the values of the gate scaling factors. The estimated values of the $\lambda_{clk}$ in each segment are used to assert a Trojan attacked or Trojan free segment of the circuit. The efficiency of proposed method is evaluated on several ISCAS benchmarks and HT detection sensitivity and accuracy is competitive with other recently published power analysis results.
5 Combinational Use of Multiple HT Detection Techniques

Combinational use of multiple Hardware Trojan (HT) detection techniques is introduced in this chapter. The effective combinational use of multiple HT detection techniques improves overall detection probability. Benchmark circuits are used to evaluate efficiency of the developed method. The discussion in this chapter is substantially drawn from [71], where we first reported the development and evaluation of this technique.

5.1 Introduction

HT detection has become critical in microelectronic circuit process due to increasing security threat. However, there is no “silver bullet” technique available yet that can be applied to detect all classes of Trojans. The majority of existing techniques address Trojan detection in manufactured ICs by a single technique that can significantly vary in detection probability for Trojans in different classes. The effective combinational use of multiple HT detection techniques can improve overall detection probability. For example, super dormant HT which is inactive in extremely long lifetime may easily invalidate HT activation strategy or in-field HT self-detection, however the HT circuit used to monitor super rare triggering signal is more complicated which result in a relatively larger size, that could be easily detected by side-channel analysis. HT circuit with small size may consume very limited timing/power that can be covered by process variation eventually invalidate HT detection based on side-channel analysis, however small HT must have simple strategy to monitor trigger signal thus HT activation technique can easily activate HT eventually benefit side-channel analysis-based HT detection.
5.2 Combinational Use of Power & Timing Analysis-based HT Detection

This section introduces combinational use of the proposed power and timing analysis-based HT detection. Benchmark circuits are used as examples to evaluate the effectiveness of the proposed technique.

5.2.1 Preliminary and Theory

The power (proposed in chapter 4) and timing (proposed in chapter 3) analysis-based HT detection methodologies are applied to circuit-under-test (CUT) with HT on different locations that account for different type of HT.

Based on location characteristics in host-circuit, HTs are classified into three categories: in-path, by-path, and off-path, as shown in Fig. 3.8. When an input pattern is applied, the location of HT plays an important role on the ability to detect it, because of its location-dependent side-channel effects.

Side-channel (timing, power) effects of HTs in various locations are listed in Table 5.1. The propagation signal in a circuit-under-test (CUT) will go through extra paths introduced by an in-path HT, that affects the measured transient time for some extent. An in-path HT may utilize the exiting signal or logic for the trigger and payload logic, so it can be simple and of minimal size, consuming little power. A by-path HT affects the CUT’s original path by introducing a capacitive load that affects the measured transient time to less extent. However, because of its limited interface with the host IC, a by-path HT may need a complicated trigger and payload design of larger size, resulting in greater power consumption. An off-path HT is isolated from the original circuit path, resulting in negligible timing alteration on the original signal propagation. An off-path HT is isolated from the original circuit, that requires more complicated trigger and payload logic to have practical effect on the host IC. Thus, an off-path HT requires an even larger circuit size with more power consumption.
### Table 5.1 Side-channel Effect of HT in Various Locations

<table>
<thead>
<tr>
<th>HT Location</th>
<th>Timing Effect</th>
<th>Power Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>In-path</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>By-path</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Off-path</td>
<td>None</td>
<td>High</td>
</tr>
</tbody>
</table>

Based on the analysis of HTs in various locations, timing and power effects of HTs on a host-circuit are trade-offs for HT designs. Therefore, a combination of power and timing-analysis will be more effective for a practical general HT detection technique. To accommodate process variation in HT detection, typically a side-channel variation limit is prescribed. A CUT is asserted as HT-attacked if its side-channel parameters fall outside of the prescribed limits. Combinational use of power and timing-based HT detection also can restrict the prescribed limits for each method, thereby increasing detection probability.

#### 5.2.2 Example

To evaluate the effectiveness of combining power-timing analysis-based HT detection, a circuit incorporating three pipelined 16-bit ripple-carry-adders (RCA) and a clock-tree is introduced as CUT, as shown in Fig. 5.1. The CUT is partitioned into three segments and clock-tree. Each partitioned segment consists of two combinational components (8-bit full adder) and one sequential gate (a D flip-flop).
The proposed timing, power, and timing-power analysis-based HT detections are applied on the CUT with Trojans, respectively. An inverter is embedded in segment 1 to account for in-path/by-path/off-path HT, as shown in Fig. 5.2. A detection circuit (green highlighted in Fig. 5.2) is attached to the HT-affected path for the proposed timing analysis. The CUT is designated as Trojan-attacked when the delay of testing-path is increased by embedding the HT. Then the proposed power-based HT detection is applied on CUT. Power profile of fabricated circuit varies with process, voltage, and

temperature (PVT) variation. The PVT variation of the clock-tree is calculated based on power measurement of each partitioned segment. HT detection in each segment is dependent on the PVT variation of the clock-tree for the given segment falling outside of the prescribed limits. Finally, the two HT detection methodologies are combined as a timing-power analysis-based HT detection technique, applying to the CUT. Based on schematic simulation results, the minimum detectable in-path HT size over host-circuit size at detection probability of 90% is 3.11% with timing analysis and 0.031% with power analysis. A HT with size of 0.027% of the host-circuit size can be detected at the probability of 37% \( (P(timing)) \) and 84% \( (P(power)) \) by timing and power analysis-based detection, respectively. While, with combinational use of the two methods, the HT detection probability is

\[
P(combination) = 1 - (1 - P(timing)) \times (1 - P(power)) = 90\%
\]

Therefore, to achieve the same detection probability, combinational use of detection methods decreases the minimum detectable HT size by 12.9%.

The full experimental results are listed in Table 5.2, in which the minimum detectable HT size over host-circuit size is at detection probability of 90%. As expected, with timing-power analysis, the minimum detectable in-path and by-path HT size are improved by 12.9% and 9.4%, respectively, but it is ineffective for off-path HTs compared to the power-analysis method.

Table 5.2 Detectable HT Size Over Host-circuit Size at 90% Detection Probability

<table>
<thead>
<tr>
<th>HT</th>
<th>Timing-based</th>
<th>Power-based</th>
<th>Timing-power-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>In-path</td>
<td>3.11%</td>
<td>0.031%</td>
<td>0.027%</td>
</tr>
<tr>
<td>By-path</td>
<td>3.86%</td>
<td>0.032%</td>
<td>0.029%</td>
</tr>
<tr>
<td>Off-path</td>
<td>N/A</td>
<td>0.032%</td>
<td>0.032%</td>
</tr>
</tbody>
</table>
5.2.3 Experimental Results

ISCAS-85 benchmarks are used to evaluate the effectiveness of combining power and timing analysis-based HT detection in general circuits. Benchmarks are modified to sequential circuit by inserting registers in path. Inverter buffers are embedded as HT. To satisfy the proposed power analysis-based HT detection, benchmarks are partitioned into four segments, in which segment 0 only consists clock-tree, and HT is randomly embedded in segment 1. A detection circuit is affiliated to the HT affected path for timing-based HT detection. Timing and power analysis-based HT detection methods are separately and combinational applied on CUT. The experimental results are listed in Table 5.3, in which “detectable HT/host-circuit size” is the minimum detectable HT size over host-circuit at detection probability of 90%. As seen in Table 5.3, timing and power-based HT detection benefits detecting in-path and by-path HT, but it is ineffective for off-path HT compared to power-based HT detection.

Table 5.3 Simulation Results of Timing & Power Analysis-based HT Detection

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Inputs/Outputs</th>
<th>Gate</th>
<th>HT Location</th>
<th>Detectable HT/Host-circuit Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Timing-analysis</td>
</tr>
<tr>
<td>S27</td>
<td>5/1</td>
<td>13</td>
<td>In-path</td>
<td>3.12%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>By-path</td>
<td>3.87%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Off-path</td>
<td>N/A</td>
</tr>
<tr>
<td>S344</td>
<td>9/11</td>
<td>175</td>
<td>In-path</td>
<td>2.54%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>By-path</td>
<td>3.31%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Off-path</td>
<td>N/A</td>
</tr>
<tr>
<td>C499</td>
<td>41/32</td>
<td>202</td>
<td>In-path</td>
<td>2.62%</td>
</tr>
</tbody>
</table>
5.3 Combinational Use of HT Activation & Side-channel-based HT Detection

Combinational use of HT activation and HT detection based on side-channel analysis is studied in this section. Benchmark circuits are used as examples to evaluate the effectiveness of the proposed technique.

5.3.1 Preliminary and Theory

Side-channel analysis is popular in HT detection, because of its high detection probability. Inserted HTs cause extra side-channel parameters consisting of timing, power, etc. HT can be revealed by measuring the differential side-channel parameters of the attacked circuit [1]. The proposed power analysis method is more accurate than timing analysis method on HT detection, because the differential power consumption of a HT-attacked circuit is easier to detect compared to differential timing. The extra delay in the HT-affected path resulting from the charge/discharge of the capacitance in the HT interface. Optimizing the size of the HT interface can effectively reduce the timing effect of HT on the host-circuit. The extra power in the HT-affected area is caused by the current paths and loads of the HT, and it is virtually independent of the location and interface design of the HT.

However, power analysis-based HT detection is not a technique with consistent high detection probability for general HTs. For example, a HT may use power gating to reduce its static current during dormant mode. The structure of power gating is shown in Fig. 5.3. The power-gated block receives its power from a virtual power rail (V_{DDV}), which is isolated from real power rail (V_{DD}) by a group of header switch transistors. If power-gated technique is used in HT design, when HT is in dormant mode the HT block
is isolated from power supply, also the off header-switch-transistors series connecting with HT block can gradually decrease circuit static power consumption because of stacking effect. Normally HT circuits are designed stealthily to avoid being detected, requiring the signal “Sleep” remains ‘1’ for the majority of the life time. Therefore, the power-gated HT is silent in both side-channel effects and functionality, which will tremendously impair most HT detection techniques.

![Diagram of Power Gating](image)

Fig. 5.3 Power Gating

Activation techniques may wake up a dormant HT to augment the HT effect on the host-chip operating parameters, facilitating detection probability of other techniques. We proposed a probability-increase-circuit (PIC) in chapter 2 to raise transition probability of rare nodes on chip to a specific level that results in the HT trigger signal occurs faster. We declare that power analysis-based HT detection together with signal activation technique will benefit HT detection probability.

### 5.3.2 Experimental Results

An experimentation was set up as follows to prove the effectiveness of combinational use of power analysis-based HT detection together with HT activation technique. A 64-bit binary comparator is used as CUT [35]. Power gated inverter buffers are randomly embedded in CUT to account for HT. The output of an 8-bit OR gate (shown in Fig. 2.4) is served as “Sleep” signal in HT power gating. All probabilities of
signals in Fig. 2.4 are labeled as format $P_0, P_1$ with assumption that primary inputs have equal probability of values. As shown, the power gated Trojan is dormant in 255/256 rate, and active in 1/256 rate. The power gated HT with sleep signal of 8-bit OR gate is shown in Fig. 5.4, in which HT is dormant (isolated from power) in the probability of 255/256, and HT output will not affect circuit original design when in dormant mode.

The power analysis-based HT detection method proposed in chapter 4 is applied on CUT with/without PIC. To satisfy the proposed power analysis-based HT detection, CUT is partitioned into four segments, in which segment 0 only consists clock-tree, and HT is randomly embedded in segment 1. Random input patterns are applied to 8-bit OR gate. The simulation results are listed in Table 5.4, in which the “Detectable HT/Host-Circuit size” is the minimum detectable HT size over host-circuit (64-bit binary comparator) size. As seen, CUT with PIC which may active embedded HT in a longer time will reduce the minimum detectable HT size from 0.73% to 0.04%.

Table 5.4 Simulation Results of Power Analysis-based HT Detection with/without HT Activation

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Inputs/Outputs</th>
<th>Gate</th>
<th>Detectable HT/Host-Circuit Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>64-Bit Binary Comparator</td>
<td>128/3</td>
<td>1314</td>
<td>0.73%</td>
</tr>
</tbody>
</table>
5.4 Conclusion

The effective combinational use of multiple HT detection techniques is introduced in this chapter. Power and timing analysis-based HT detections are combined to apply to a benchmark circuit and show a distinct improvement in detecting in-path and by-path HTs, but it is ineffective for off-path HTs compared to power-based HT detection. Subsequently, side-channel analysis-based HT detection was applied together with an HT activation that improved detection of all classes of HTs.
6 Dynamic CMOS Circuit Design Optimization

CMOS circuit design optimization is considered for optimizing parametric overhead of HT detection circuit. A dynamic CMOS circuit design optimization technique is proposed. Some general used basic components (AND gate, NAND gate, full adder) are applied to prove the effectiveness of the proposed technique. The basic idea of this technique is that avoidance of discharging pull-down-network (PDN) nodes during the evaluation mode is effective in optimizing power-delay-product (PDP) of dynamic CMOS circuits. The discussion in this chapter is substantially drawn from [72], where we first reported the development and evaluation of this technique.

6.1 Introduction

Circuit operation at high frequencies while consuming low power is one of the most important characteristics in designing integrated circuits (ICs). Compared with static CMOS circuits, dynamic CMOS circuits are faster because they have lower load capacitance, no contention during switching, and no static power dissipation; however dynamic circuits have higher power consumption due to the operating mechanism. The high speed of dynamic circuits resulted in this class of circuits having an important role in the high performance digital IC market. But in recent years, the power hungry dynamic ICs are becoming a heavy load for battery and heat dissipation demands in portable IC products [73][74].

The demand for high speed coupled with low power (especially in small portable applications) provides motivation for improvement of PDP for dynamic CMOS circuits. Two techniques are proposed for modifying the conventional CMOS dynamic circuit which significantly benefit circuit PDP. The techniques can simply modify existing dynamic CMOS designs with virtually no effect on layout area. Conventional
benchmark circuits and the modified circuits using the proposed techniques are implemented in IBM 90nm CMOS technology with a 1.2V power supply. Simulation results indicate that the proposed techniques can improve circuit PDP by 19.2% and 61.9% in two non-inverted dynamic benchmarks, respectively, and 6.2% and 33.72% in two inverted dynamic benchmarks, respectively.

6.2 Conventional Dynamic CMOS Circuit

![Conventional Dynamic CMOS Circuit Diagram](image)

Fig. 6.1 Conventional Dynamic CMOS Circuit, (a) Non-Inverted, (b) Inverted, (c) Operation Modes

As seen in Fig. 6.1 (a), a conventional non-inverted dynamic circuit has two modes of operation, pre-charge and evaluation, shown in Fig. 6.1 (c). The circuit initializes in the pre-charge mode, then functions in the evaluation mode. During pre-charge mode, the signal clock is low, the clocked PMOS M1 is on and the output is initialized to a high voltage. In evaluation mode, the clock is high, the clocked PMOS M1 is turned off, so output is discharged to low through foot transistor M2 if the PDN
is on, otherwise output remains at a high voltage. The conventional inverted dynamic circuit, shown in Fig. 6.1 (b), initializes to a low voltage output in pre-charge mode, and functions in evaluation mode. Therefore, PDN is off in pre-charge by having the low dynamic circuit output as inputs, so the foot NMOS transistor is unnecessary.

Dynamic circuits are vulnerable to input noise and leakage due to floating nodes. Unwanted noise on an input which intermittently exceeds threshold voltage $V_t$ may result in the output slowly discharging without recovering. Also, when an input node floats to high voltage during evaluation, the output node is slowly discharged due to subthreshold gate and junction leakage, again with no recovery to the desired final output. To overcome the noise margin and leakage problems, a voltage keeper (M3) is connected to output pulling up voltage when necessary, as seen in Fig. 6.1 [75].

### 6.3 Dynamic CMOS Circuit Design Optimization Techniques

A key shortfall with the conventional circuits is the power and delay increase associated with discharging of nodes in the PDN which were coincidently set to a high voltage while taking the output node to a high voltage. Discharging of the voltage on nodes of the PDN also results in increase power consumption in taking the output low while contending with the voltage keeper attempts to maintain a high voltage.

The proposed techniques are based on discharging PDN nodes and avoiding charging PDN nodes in pre-charge mode, which eliminates the need to discharge PDN nodes during the evaluation mode. This section will discuss the two techniques and the resulting performance improvements.

#### 6.3.1 Proposed Non-Inverted Dynamic CMOS Circuits

During the pre-charge mode for the conventional non-inverted dynamic circuit design, shown in Fig. 6.1 (a), all PDN nodes are charged to $Vdd$ while having dynamic outputs as inputs. The proposed technique moves the foot NMOS transistor to the top
of NMOS stack, which is denoted as the up-footed non-inverted dynamic circuit. A dynamic 2-input NAND gate, shown in Fig. 6.2, is used to illustrate the improvements obtained with the up-footed non-inverted dynamic circuit design.

Fig. 6.2 Dynamic 2-Input NAND Gate, (a) Conventional, (b) Up-Footed

As shown in Fig. 6.2 (a), in typical dynamic NAND gate, the signal clock and all inputs are ‘1’ in pre-charge, and all PDN nodes (N2, N3) are charged. Therefore, if the output (Y) needs to go low in evaluation, the circuit needs to discharge the output node (N1) and also the PDN nodes (N2, N3) resulting in increased PDP. Because 1) the unnecessary charging of PDN nodes consumes more power in pre-charge; 2) discharging PDN nodes takes more timing while pulling down output node in evaluation; 3) contention time of PDN pulling down output node and voltage keeper maintaining high voltage is increased in evaluation, resulting in more timing and power consumption. However, in the up-footed dynamic NAND gate, shown in Fig. 6.2 (b), the up-foot M2 transistor isolates the PDN from voltage high in pre-charge mode, eliminating any pre-charging of nodes in the PDN.

6.3.2 Proposed Inverted Dynamic CMOS Circuits

In the conventional inverted dynamic circuit design, shown in Fig. 6.1 (b), if
dynamic inputs remain ‘0’ in pre-charge, the PDN node(s) are isolated from the power source high voltage. However, if continuous input patterns result in charging of PDN nodes in one evaluation, then discharging these nodes in succeeding evaluation mode continues to be an issue. If the probability of inputs being high is known, the lower high-voltage-probability input should place at the top of PDN to minimize unnecessary node charging. Otherwise, we propose a clock controlled NMOS transistor to discharge the PDN nodes in the pre-charge mode. The NMOS transistor used for discharging is denoted as node-discharger. A dynamic 2-input AND gate, shown in Fig. 6.3, is used to illustrate the operation of the node-discharger. Fig. 6.4 (a) shows the expected ideal waveforms for both conventional dynamic AND gate and dynamic AND gate with node-discharger. Fig. 6.4 (b) shows the Cadence schematic simulation waveform results.

Fig. 6.3 Dynamic 2-Input AND Gate (a) without (b) with Node-Discharger

For conventional dynamic AND gate, shown in Fig. 6.3 (a), node N1 is pulled up to high voltage in pre-charge mode. Input “AB” is “10” in the following evaluation, then NMOS M2 is on, M3 is off, and node N2 is charged (at time t1 shown in Fig. 6.4 (a)). In the succeeding evaluation, input “AB” is “11”, then the PDN needs to pull down both nodes N1 and N2 resulting in increased PDP. However, in modified dynamic AND
gate, shown in Fig. 6.3 (b), the node-discharger (M5) can discharge node N2' in succeeding pre-charge mode (at time t2 shown in Fig. 6.4 (a)), then regardless of previous input vectors, N1’ is always the only node that may need discharging. Simulation results in Fig. 6.4 (b) show the delay at the output is reduced 2.34ps by adding node-discharger.

![Diagram](image)

Fig. 6.4 Operation Waveform of Dynamic 2-Input AND Gate, (a) Ideal (b) Simulation

Although node-discharger can save power consumption during the evaluation mode, node-discharger itself may consume extra power during the pre-charge mode. The node-discharger has the entire pre-charge mode to discharge nodes, so timing is non-critical concern for discharging. Therefore, in order to optimize area and power overhead caused by node-discharger, a minimum transistor size is used. Based on
simulation results, for some designs, the circuit with node-discharger may consume more power compared with conventional circuit, but the PDP still benefits from this technique.

6.4 Final Design Performance

The benchmarks we use to validate the proposed algorithm are dynamic 3-input NAND gate, 3-input AND gate, and 1-bit full adder [76][77]. Circuits are implemented in IBM 90nm CMOS technology with a 1.2 V power supply. The widths of pulldown transistors are chosen to have unit resistance. Pulling up output takes place in pre-charge mode while timing is out of concern. Then PMOS is with width to have twice unit resistance that reduces the load resistance of signal clock.

The performance comparison of conventional non-inverted dynamic reference circuits and modified circuits is shown in Table 6.1, where the “improvement” row is for the improvement of up-footed circuit compared with conventional circuit. Input pattern changes in every clock cycle, and all input cases are used for simulations.

As seen in Table 6.1, circuit delay and power consumption are improved in up-footed design, in which PDP is reduced by 19.2% and 61.9% for NAND3 and 1-bit full adder, respectively.

Table 6.1 Performance Improvement in Non-Inverted Dynamic Benchmarks

<table>
<thead>
<tr>
<th>Circuit</th>
<th>3-Input NAND Gate</th>
<th>1-Bit Full Adder</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Conventional</td>
<td>Up-Foot</td>
</tr>
<tr>
<td>Delay (ps)</td>
<td>114.93</td>
<td>104.17</td>
</tr>
<tr>
<td>Power (μW)</td>
<td>0.906</td>
<td>0.804</td>
</tr>
<tr>
<td>PDP (fJ)</td>
<td>0.104</td>
<td>0.084</td>
</tr>
<tr>
<td>Improvement</td>
<td>9.36%</td>
<td>11.3%</td>
</tr>
<tr>
<td></td>
<td>19.2%</td>
<td>61.9%</td>
</tr>
<tr>
<td>Conventional</td>
<td>66.59</td>
<td>35.81</td>
</tr>
<tr>
<td>1-Bit Full Adder</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Up-Foot</td>
<td>7.086</td>
<td>5.035</td>
</tr>
<tr>
<td>Improvement</td>
<td>46.2%</td>
<td>28.9%</td>
</tr>
<tr>
<td></td>
<td>61.9%</td>
<td></td>
</tr>
</tbody>
</table>
The performance comparison of conventional inverted dynamic reference circuits and modified circuits is shown in Table 6.2, in which “improvement” row is for the improvement of circuit with node-discharger compared with conventional circuit. Input patterns are set up to charge PDN nodes as much as possible in one evaluation, then pull down output in the following evaluation, in which case node-discharger improves the circuit performance the most.

As seen in Table 6.2, circuit delays are reduced with node-discharger, but the AND gate consumes more power compared to conventional design. Because charging node-discharger NMOS consumes extra power during the pre-charge mode while charging the nodes in PDN. However, PDP of the two circuits with node-discharger are all reduced.

<table>
<thead>
<tr>
<th>Table 6.2 Performance Improvement in Inverted Dynamic Benchmarks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>3-Input AND Gate</strong></td>
</tr>
<tr>
<td>Conventional</td>
</tr>
<tr>
<td>Node-Discharger</td>
</tr>
<tr>
<td>Improvement</td>
</tr>
<tr>
<td><strong>1-Bit Full Adder</strong></td>
</tr>
<tr>
<td>Conventional</td>
</tr>
<tr>
<td>Node-Discharger</td>
</tr>
<tr>
<td>Improvement</td>
</tr>
</tbody>
</table>

**6.5 Conclusion**

This chapter presented two design techniques for improving performance of dynamic CMOS circuits: up-footed design by moving foot NMOS to top of NMOS PDN stack and node-discharger design by adding node-discharger to node(s) in PDN.
They both benefit circuit PDP by eliminating the need to discharge PDN nodes during the evaluation mode. Benchmark circuits, dynamic NAND3/AND3 gates and dynamic 1-bit full adder, are used to validate the techniques. Cadence simulation results with 90nm CMOS technology indicate that for non-inverted dynamic circuit, the PDP is reduced by 19.2% and 61.9% in two benchmarks, respectively. For the inverted dynamic circuit, delay is always improved, but power consumption may increase for some designs; however, the PDP improves by 6.2% and 33.72%, respectively for the benchmark circuits.
7 Conclusion and Future Work

7.1 Conclusion

This dissertation has focused on contribution to the field of hardware Trojan (HT) detection and CMOS circuit design optimization that optimizes overhead of HT detection circuit in host-chip.

7.1.1 Hardware Security

A general study of hardware security has been presented. Three HT detection methodologies which are based on HT activation and side-channel analysis are proposed.

7.1.1.1 HT Detection Efficiency Improvement

Due to stealthy nature, typically HT is triggered by a rare signal and remain dormant mode in most lifetime, that lagging HT detection period. A signal probability-increase-circuit (PIC) is proposed inserting to nodes with low transition probability for reducing transition time. Afterwards all nodes on host-chip could have transition probability more than a specific level, and rare signals in all nodes could appear in greater opportunity. Activation time of rare signal-triggered HT will be tremendously decreased, that making the HT detection more efficient. Based on simulation results, trigger time of rare signal in CUT is tremendously reduce with modest circuit overhead.

7.1.1.2 Timing Analysis-based Hardware Trojan Detection

An HT circuit embedded in testing-path will add extra capacitance, resulting in more charging and discharging delays to the testing-path. Then the delay time will be increased for a HT-attacked CUT compared to HT-free CUT due to the additional delay caused by HT circuit. This dissertation proposes a timing analysis-based HT detection technique detecting HT by revealing the timing deviation.
Compared to the state-of-the-art timing analysis-based detection, the main contributions of the proposed technique are as follows:

- HT detection circuit area, timing, and power overhead on host-circuit are reduced.
- HT detection technique is not restricted to be applications with specific paths, but on any circuit path by isolating the path with extra registers.
- Location of HT can be estimated.
- Tunable detection probability to accommodate different security requirements of circuit designs.

Based on simulation results, the ratio of detectable Hardware Trojan size over host-circuit size ranging from 2.81% to 3.37% at 90% detection probability. For FPGA implementation of HT detection, the ratio of detectable HT size to host-circuit size ranging from approximated 0.5% to 0.9% at 90% detection probability. The detectable Trojan size and detection probability can be further improved by applying more detection circuits on testing-path. Also, ring-oscillator can be introduced to estimate operation temperature and process variation of testing-path to calibrate detection parameters, eventually improve detection probability.

7.1.1.3 Self-Reference-based Hardware Trojan Detection

Embedded HT will add current paths and loads to the original circuit, that result in extra power consumption on wires and gates in HT affected area. A self-reference-based HT detection method is proposed revealing HT by measuring the power deviation of the attacked circuit.

The main contributions of this technique are as follows:

- HT detection method can be applied with zero-overhead.
- Genuine chip is not needed.
- Location of HT can be estimated.
- Partitioned CUT increases detection accuracy.
- Detection accuracy is adjustable by size of partitioned segments.
- Computational complexity is optimizable based on required detection accuracy.

The efficiency of proposed method is evaluated on several ISCAS benchmarks and HT detection sensitivity and accuracy is competitive with other recently published power analysis results.

### 7.1.1.4 Combinational Use of Multiple HT Detection Techniques

The proper combinational use of multiple HT detection techniques can benefit detection probability of each technique, eventually it will be competitive than application of any single detection technique.

A HT detection method may be effective for HT in some classes. Multiple HT will cover more kinds of HT. To accommodate process variation in HT detection, typically a side-channel variation limit is prescribed to reveal extra side-channel parameters caused by HT. CUT is asserted as HT-attacked if its side-channel parameter falls outside of the prescribed limits. Combinational use of multiple side-channel analysis-based HT detection also can restrict the prescribed limits for each other, eventually also facilitates detection probability.

Benchmark circuits are used to evaluate efficiency of the developed method, and the proposed technique has distinct improvement in detecting HT.

### 7.1.2 CMOS Circuit Design Optimization

CMOS circuit design optimization is considered for optimizing parametric overhead of HT detection circuit.

The proposed dynamic CMOS circuit design optimization is based on discharging pull-down-network (PDN) nodes and avoiding charging PDN nodes in pre-charge
mode, which eliminates the need to discharge PDN nodes during the \textit{evaluation} mode. Simulation results indicate that for non-inverted dynamic circuit, the PDP is reduced by 19.2\% and 61.9\% in two benchmarks, respectively. For the inverted dynamic circuit, delay is always improved, but power consumption may increase for some designs; however, the PDP improves by 6.2\% and 33.72\%, respectively for the benchmark circuits.

7.2 Future Work

7.2.1 Hardware Security

With parameter-muted design (with power gating, etc.), HT may bypass all pre-deployment detections. A self-detection design is required in field to compensate this loophole. Self-detection circuits should have three specifications: 1) low overhead, 2) work friendly with host-chip at speed, 3) react with HT to stop hurting chip-user (turn off power, etc.). The study of this detection will be the future work.

7.2.2 CMOS Circuit Design Optimization

Power consumption of IC consists of dynamic power and static power. Dynamic power is consumed by switching signals, while static power is consumed by leakage current when gate is not switching. Although static power is negligible compared to dynamic power, accumulation of million gates on chip and the circuits in sleeping mode make static power a primary concern. Static power is proportional to static leakage current which consists of gate leakage, subthreshold leakage and junction leakage \cite{78}. Gate leakage is caused by thin gate oxides when a voltage is applied on gate that may electronically tunnel from gate to body. Subthreshold leakage occurs when a transistor is off that supposed to have zero current. Junction leakage current flows when source or drain diffusion region is at a different potential from the substrate. The study of circuit design optimizing these static power consumptions will be the future work. A
low static power CMOS gate design technique will benefit both dynamic and static CMOS circuit.
8 Reference


