Chapter X: Detecting Hardware Trojans using Delay Analysis

Jim Plusquellic - University of New Mexico (jimp@ece.unm.edu)
Fareena Saqib - University of North Carolina Charlotte (fsaqib@uncc.edu)

Abstract

The migration of the semiconductor industry to a distributed and out-sourced business model, in which design, integration, manufacturing, packaging, test and assembly are activities carried out by multiple companies all over the world, has created challenges related to intellectual property (IP) piracy, reverse engineering attacks on netlists and chip layouts, integrated circuit (IC) cloning and detecting counterfeit chips and hardware Trojans (HT). In particular, threats created by the insertion of hardware Trojans (HT) have emerged as a serious concern among commercial vendors, government agencies and their contractors. A wide variety of HT detection methods have been proposed, including those based on logic and parametric (delay, power, thermal, etc.) testing strategies, as well as those that destructively validate images of the chip layout against the trusted design data. This chapter surveys path-delay-based parametric approaches for detecting HT. The primary benefits of this approach over other methods is the non-destructive nature of the testing and the high sensitivity that such methods provide to detecting small analog anomalies introduced by HT.

1. Introduction

Hardware Trojans (HT) are a deliberate and malicious change to an IC that adds or removes functionality or reduces reliability of an integrated circuit (IC) or system [1-8]. The changes can be designed to leak secret information, e.g., encryption keys or other types of private internal information, or they may be designed to cause the system to fail at some specific or predetermined time while the IC is in mission mode. Adversaries design the HT to be difficult to discover, either accidentally via manufacturing test or purposely using tests specifically designed to activate the HT. Sophisticated HT insertion strategies also consider resilience to advanced HT detection methods that utilize high resolution measurements of *side-channel* signals, such as electromagnetic (EM) emanations, power consumption (steady-state I_{DDQ} or transient I_{DDT}), delay testing and temperature profiling.

In addition to these testing challenges, HT detection methods are further tasked to deal with several other fundamental HT properties. First, the task of identifying an HT is akin to finding a needle in a haystack, i.e., the adversary has a huge advantage because he/she can choose to insert the HT anywhere while the trusted authority is tasked with determining if the IC has in fact been modified and if so, finding the unknown malicious function in a 'sea' of gates. Second, HT and 'bugs', either hardware or software, share the same characteristics, and it is widely accepted that finding all the bugs in a complex program is generally infeasible [9]. In fact, cleverly inserted HT can be designed to appear as bugs, making it difficult to decide if the malicious function, if discovered, was accidental or purposeful. Third, any attempt by the trusted authority to increase the 'ease' of HT detection may be visible to the adversary, i.e., the adversary can reverse engineer the IC and avoid countermeasures added by the trusted authority. Fourth, the adversary can choose to 'selectively' insert the HT into only a subset of the manufactured ICs, making it necessary to verify all manufactured ICs. Last, HT designed to leak information may not cause a change in the functional behavior of the IC and therefore, the trusted authority may need to apply non-standard tests, e.g., tests for anomalous EM radiations. Moreover, the appropriate detection strategy will

vary greatly depending on the assumptions made regarding the “insertion point”, i.e., design-inserted HT require very different detection techniques than those inserted into a layout description of the design.

The only advantage afforded to the trusted authority is that his/her detection strategy can be ‘parallelized’ because the HT needs to be detected only once and is, in most cases because of mask cost issues, inserted in the same fashion in every copy of the targeted IC population. Therefore, tests applied post-manufacturing can be partitioned among multiple independent IC testers (referred to as automatic test equipment or ATE), and applied in parallel. Unfortunately, even high levels of parallelism ‘run-out-of-gas’ when the full extent of the search space, both combinational and sequential, is considered.

This chapter is focused on surveying methods that utilize very precise analog-based testing to discover HT. The underlying basis of these methods can be characterized by the Heisenberg principle or *observer effect*, i.e., any attempt to measure or monitor a system changes its behavior. The testing methods described herein attempt to determine if an adversary has inserted an HT that is ‘observing’ the evolving state of the IC, which is used by the adversary as the mechanism to activate the HT. In particular, we survey path delay-based testing methods which are designed to detect subtle changes in delay introduced by the HT connections and gate insertions, referred to as the trigger and payload of the HT, respectively. The authors of [10] propose a generic characterization of these concepts as shown in Fig. 1.

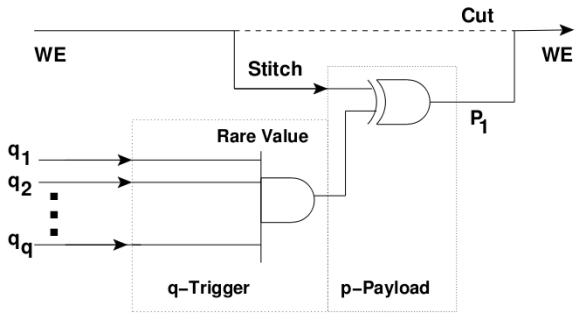


Fig. 1. Generic characterization of a Hardware Trojan trigger and payload from [10]. Trigger signals q_1 through q_q typically connect to nodes in the existing design and therefore add capacitive load to these signals, creating an *observer effect*. Both the trigger signals and payload add delay to paths in the existing design.

The rest of this chapter is organized as follows. Section 2 presents a high level view of HT insertion strategies, and discusses the constraints on the detection methods. Section 3 covers HT detection strategies designed to detect layout or GDSII Trojans (other HT insertions points are detailed in other chapters of this book) with subsections that survey detection methods that analyze ‘side-channel’ signals, e.g., power and delay. Section 4 describes important fundamental concepts related to implementing path delay-based HT methods. Section 5 provides a survey of delay-based HT detection techniques, while Section 6 describes the first proposed multiple-parameter side-channel method. Conclusions are provided in Section 7.

2. Hardware Trojan Insertion Points

The horizontal dissemination of the IC design, fabrication and test processes to many distinct companies around the world has dramatically increased the potential for malicious activities. Intellectual property (IP) block reuse has compounded this threat by partitioning the design space itself among multiple third party vendors. Standardization activities have enabled multiple independently designed IP blocks to seamlessly integrate into CAD tool flows. However, the elec-

tronic design automation (EDA) community developed this multi-party collaborative design system using a model in which all parties are largely trusted. Unfortunately, the same types of malicious activities endured by the software community are now presenting themselves in the hardware design community.

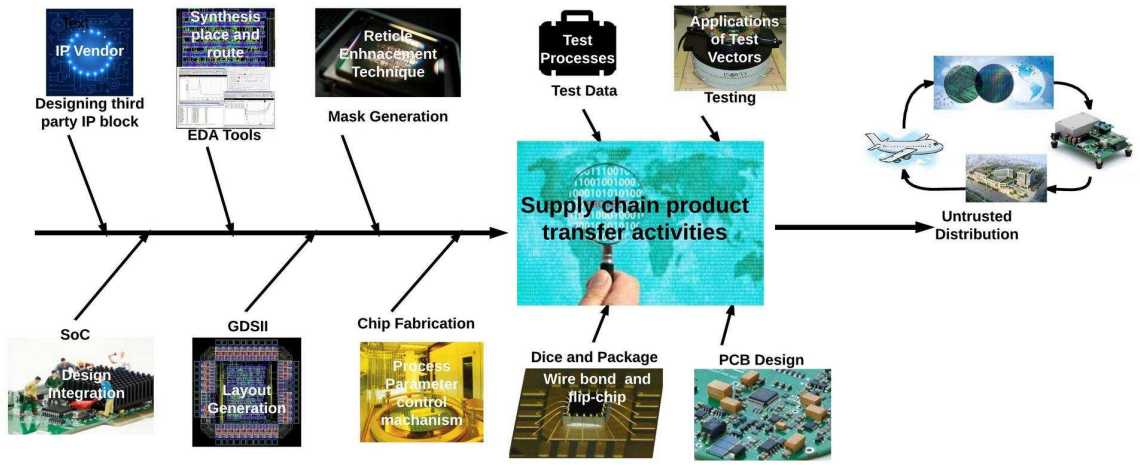


Fig. 2. Hardware Trojan insertion points.

All of the primary processes associated with design, manufacture and test are vulnerable to malicious activities where adversaries can add to, remove from or change the functionality of the IC. We refer to these opportunities as **insertion points**. Fig. 2 provides a graphical illustration of the major insertion points, which are further distinguished by the following list:

- Designing third party IP blocks
- Developing CAD tool scripts
- Integration activities where IP blocks and glue logic are assembled into System-on-Chip (SoC) ICs
- Behavior synthesis and place and route (PnR) carried out by CAD tools
- Layout mask data generation and mask preparation
- Process parameter control mechanism used in the multi-step fabrication process
- Supply chain transactions associated with transferring wafers from one facility to another
- Generating test vectors using automatic test pattern generation (ATPG)
- Wafer-probing activities associated with measuring test structures and detecting defects
- Supply chain transactions associated with creating and transferring dice
- Processes responsible for packaging ICs
- Applying ATPG vectors to packaged ICs using ATE
- Supply chain transactions associated with transferring packaged parts
- Printed circuit board (PCB) design and fabrication
- Processes responsible for installing PCB components (populating PCBs)
- Supply chain transactions associated with transferring boards
- System integration and deployment activities

The wide range and widely distributed nature of these activities presents an overwhelming opportunity for subversion. Moreover, the wide diversity among the tasks will require a very sophisticated and complex system to manage the entire set of trust vulnerabilities from start to finish. The research community is tackling the trust challenges one-at-a-time, and is focused on those that are the most attractive insertion points for adversaries. For example, subversion of IP blocks is a serious concern given the ease in which malicious functionalities can be covertly

inserted and the absence of alternate representations and models to which the IPs can be compared [11]. Layout modifications and IC fabrication insertion points represent another important focus area, especially given the huge complexity associated with analyzing fabricated ICs at this lowest layer of design abstraction, and the wide range of opportunities available to the adversary in designing HT with sophisticated, sometime analog, triggering and payload mechanisms.

Note that significant differences exist in the HT countermeasures and detection strategies that are applicable even when only considering these two insertion points. For example, golden models are not available at the IP block insertion point, but architectural changes that obfuscate the design are available as countermeasures. On the other hand, the layout insertion point allows layout design data to be used to validate the functional and analog behaviors of the IC, but obfuscation is limited to 'dummy via' insertion and other nano-level manipulations of the design. Also, side channel information is not available or is not accurate enough to be useful for IP blocks but can be leveraged as a very powerful HT detection method for layout-level validation. The focus of this chapter is on HT detection methods, and countermeasures where appropriate, that are applicable at the layout level. Other chapters of this text survey techniques which target other insertion points.

3. Approaches to Detecting Layout-Inserted Hardware Trojans

A layout is a physical representation of the design, i.e., it is a set of geometric shapes that represent a physical model of the IC. The shapes define transistors, wires, vias and contacts. A layout is the lowest layer of abstraction in the design process and contains all the logic gates that define the function as well as all of the electrical connections between the logic gates and the power supply rails. The complexity of layouts increases as technology feature sizes shrink into the nanometer regime and additional wiring layers are added. Fig. 3 shows several standard (std.) cell layouts on the left and a tool synthesized layout of a relatively small functional unit called the Advanced Encryption Standard (AES). The layout of the AES IP block contains approx. 12,000 std. cells and 50,000 wires and typically would represent one IP block of several 100 on a modern SoC. The technology used in this example is an IBM 90 nm process which provides nine vertically-stacked layers for metal wires. The image is a screen capture of the designer's view of the layout using CADENCE Virtuoso. Most layout design tools provide this type of top-down view, with upper metal layers obscuring the transistors in the bottom-most layer, i.e., nearly all of what is shown in the AES layout are metal wires.

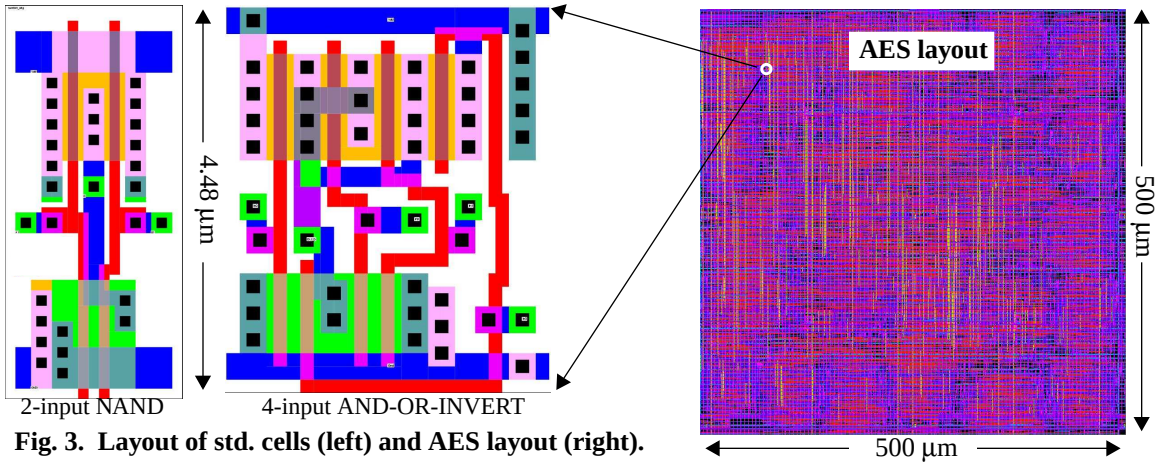


Fig. 3. Layout of std. cells (left) and AES layout (right).

Once the physical model of the layout is completed as shown by the AES layout in Fig. 3, a set of masks are generated. The masks decompose the layout into a set of (x,y) planes, which can

be vertically aligned to define the transistors and wiring layers. Layout-inserted HT are characterized as changes in one or more of the masks used in photolithography process to create physical instances of the IC. The multitude of overlapping wires and the tightly packed form of the transistors define a complex structure which represents the haystack in the 'needle-in-a-haystack' paradigm. Adversaries are free to add or change very small regions in the masks, which can effect connectivity relationships between a small set of existing std. cells, or new std. cells can be added. The latter is possible using 'white space', i.e., areas in the lowest layers of the layout that contain non-functional filler cells or cells implementing decoupling capacitors.

3.1 Layout-Oriented HT Detection Methods

HT detection methods which are designed to detect malicious modifications to the IC layout fall into three fundamental categories [1]:

- Non-destructive logic-based testing methods
- Non-destructive side-channel-based testing methods
- Destructive physical inspection techniques

3.1.1 Non-Destructive Logic-Based HT Detection Methods

Logic-based methods derive test vectors that attempt to activate the HT [12-18]. Unlike manufacturing tests which activate and propagate faults on each node individually within the fabricated IC, HT activation is akin to multiple fault activation, which is rarely practiced in manufacturing test because of the high time complexity for ATPG and high cost of applying very large numbers of vectors. Also, unlike manufacturing defects which tend to distribute randomly across circuit nodes, the adversary chooses a stealthy location for the HT, i.e., he/she inserts the HT on circuit nodes that are difficult to control or observe. Unfortunately, the task of generating test vectors that provide coverage of all possible states for these nodes is orders of magnitude more difficult than it is for manufacturing defects, and therefore, achieving high levels of HT coverage is difficult or impossible given limited resources and existing manufacturing test cost-constraints. The authors of [12-18] present alternative test generation strategies that are optimized to deal with these challenges, either alone or in combination with design modifications and side-channel-based testing approaches, as detailed in other chapters of this text.

3.1.2 Side-Channel Analysis Approaches

Side-channels refer to access and measurement techniques that bypass the designer-intended input-output mechanisms, e.g., the digital I/O pins of an IC. Side-channels, as the name implies, refer to auxiliary electrical and/or electromagnetic (EM) access mechanisms, such as the V_{DD} and GND (power supply) pins or the top-layer metal connections in the physical layout of the IC. Side-channel attacks utilize these auxiliary electrical paths to introduce signals, usually in an attempt to create a fault while the IC is operational [19], or to measure signals, in an attempt to extract private internal information [20].

Side-channels can also be leveraged by the trusted authority to obtain information regarding the integrity of the IC. For example, leakage current (I_{DDQ}) and transient current (I_{DDT}) measurements have been widely used to detect manufacturing defects [21-23]. Moreover, the trusted authority can also introduce on-chip design-for-testability (DFT) [24] and other types of specialized instruments [25][26] which allow access to additional side-channels that are not directly accessible using auxiliary channels to the IC. DFT components are designed to improve visibility of the internal and localized behavior of the IC, and include mechanisms to measure path delays, localized transients and temperature profiles. DFT added by the trusted authority can also be leveraged by adversaries as 'backdoor' access mechanisms to internal secrets, e.g., encryption

keys, so security features such as fuses must be included to disable DFT after the IC is fabricated.

Side-channel signals are typically *analog* in nature, and can provide detailed, high resolution information about the internal timing and regional signal behavior of the IC. For example, I_{DDT} measurements reflect performance characteristics of individual gates as logic signals propagate along one or more paths in the circuit. This type of temporal information can be reverse-engineered and compared with simulation-generated data to validate the structural characteristics of the fabricated layout, i.e., to ensure the chip is consistent with the ***golden model*** representation described by the design data.

Path delay measurements, if measured at high resolutions, can also serve this role. Unlike I_{DDx} measurements which provide a large-area regional observation, path delays are influenced by only those components that interact with the wires and gates along the ***sensitized*** path (defined as a path that propagates a logic signal transition). Therefore, path delay measurements can potentially be used to define a high resolution HT detection methodology. Unfortunately, path delays are also affected by variations which occur in fabrication processing conditions, commonly referred to as ***process variations***. Path delay variations caused by process variation effects are unavoidable and must be distinguished from delay variations introduced by an HT. Failure to do so is costly both in terms of the time and effort involved in verifying false alarms and, worse, in HT escapes that leave fielded systems vulnerable to attack. Subsequent sections of this chapter investigate both the benefits and challenges of using path delays as a HT detection method.

3.1.3 Destructive Physical Inspection-Oriented Methods

A third tactic to determining whether a chip is free of malicious inclusions is to apply destructive delayering and imaging techniques. Companies such as Tech Insights [27] and Analytical Solutions [28] provide services that reverse-engineer the physical characteristics of a chip to design data such as a schematic, which can then be inspected to identify IP infringements or HT circuitry. Failure analysis techniques including scanning optical microscopy (SOM), scanning electron microscopy (SEM), pico-second imaging circuit analysis (PICA), voltage contrast imaging (VCI), light-induced voltage alternation (LIVA), charge-induced voltage alternation CIVA, are used as needed in the reverse engineering process [1][29]. The primary disadvantage of these methods is their high cost and long processing times. Moreover, many destroy the chip, and therefore, cannot be used to validate chips for field use.

4. Fundamentals of Delay-Based HT Detection Methods

This section introduces the three fundamental technical domains that need to be considered by path delay-based methodologies: 1) the test vector generation strategy, 2) the technique employed for measuring path delays, and 3) the statistical detection method for distinguishing between process variation effects and HT anomalies. A commercially viable HT detection method must address each of these in a cost-effective manner. We investigate the challenges associated with each of these domains and describe proposed solutions in this section. Many of the methods surveyed in Section 5 address only a subset of these technical domains and therefore must be combined with other techniques to be fully operational in practice.

4.1 Path Delay Measurement Schemes and Other Concepts

When technology scaling entered the deep submicron era circa 2000, higher frequency operation, within-die variations, coupling, modeling challenges and power supply noise drove the IC design and test community to more sophisticated statistical modeling approaches for IC development and test [30-31]. This era also renewed interest in delay fault models [32], namely transition fault, gate delay fault and path delay fault models, which were introduced earlier in previous

works [33-36]. Although it became apparent that delay fault testing was needed to keep defect levels low, work-arounds were developed to allow the 2-vector sequences which define a delay fault test (described below) to be applied. The work-arounds became known as *launch-on-shift* (LOS) and *launch-on-capture* (LOC). LOS and LOC allow 2-vector delay tests to be applied while minimizing the amount of additional on-chip logic needed to support this type of manufacturing test.

Unfortunately, LOS and LOC delay test mechanisms also create constraints on the form of the 2-vector sequences, i.e., they do not allow the 2 vectors that define a sequence to be independently specified. These constraints reduce the level of fault coverage that can be attained for delay defects. More elaborate design-for-testability (DFT) structures that do allow both vectors of the sequence to be independently specified have been proposed [24], but are difficult to justify because of their negative impact on area and performance, and the fact that they would only be used during manufacturing test. These constraints continue to hold for modern day SoCs. However, increasing awareness of hardware trust concerns may provide the impetus for a paradigm shift which would justify additional on chip support, particularly given the significant security and trust benefits associated with path delay testing, as we discuss in the following.

4.1.1 Path Delay Testing Defined

Path delay tests are defined as a 2-vector sequence $<V_1,V_2>$, with the initialization vector V_1 applied to the inputs of a circuit at time t_0. The circuit is allowed to stabilize under V_1. At time t_1, vector V_2 is applied and the outputs are *sampled* at time t_2. The *Clk* signal is used to drive both the launch flip-flops (FFs), which apply V_1 and V_2 to the combinational block inputs, and the capture FFs which sample the new functional values produced by V_2. The time interval (t_2 - t_1) is referred to as the ***launch-capture interval*** (LCI), and is typically set to the operational clock period for the chip. Fig. 4 shows the standard form of a path delay test. Note that the standard form places no constraints on the values used for V_1 and V_2.

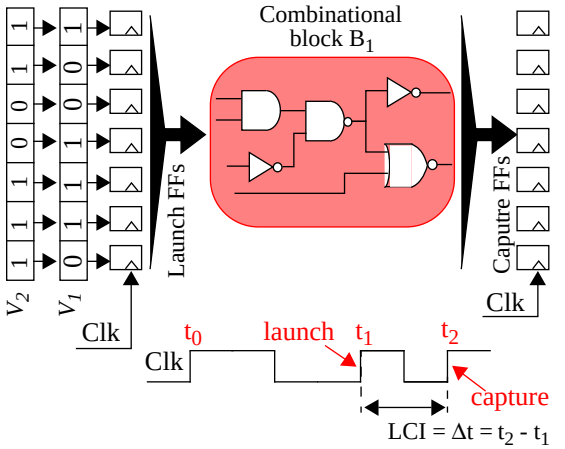


Fig. 4. Standard form of path delay test.

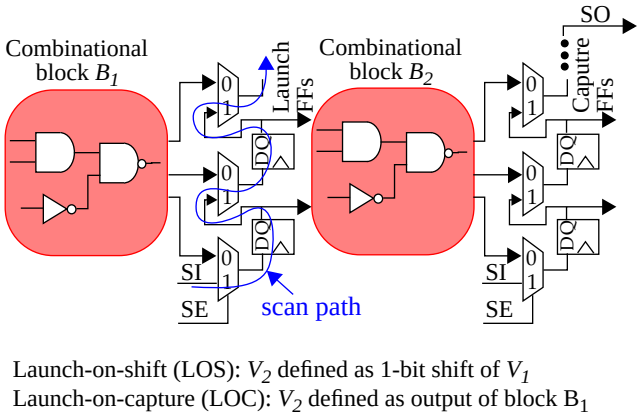


Fig. 5. Actual form uses scan flip-flops.

Unfortunately, external, off-chip access to the Launch and Capture FFs which connect to the combinational blocks within an IC is not possible. Fig. 5 shows a typical configuration with several cascaded combinational blocks B_1 and B_2, with interleaved FFs. The manufacturing test community introduced a design-for-testability (DFT) feature called **scan** to address the difficulty of applying manufacturing tests to embedded combinational blocks [24]. Scan provides a second, serial path through all (or most) of the FFs in the IC. The second path is commonly implemented

by adding a 2-to-1 MUX before the input of every FF (as shown in the figure). A *scan-enable* (SE) control signal is added as an I/O pin on the chip to allow test engineers to enable the serial path, and to shift in a sequence of 0's and 1's using a second I/O pin referred to as *scan-in* (SI). The scan path allows the internal FFs to be configured with test data that is designed to maximize fault coverage. Once a test vector is scanned into the chip, *Clk* is used to apply a launch-capture test, which captures the functional outputs of the blocks B_i in the Capture FFs. A second scan operation allows those values to be read out using a third I/O pin called *scan-out* (SO).

The scan architecture shown in Fig. 5 allows only a single vector V_1 to be applied. Manufacturing tests that target defects which prevent circuit nodes from switching (called stuck-at faults) can be applied directly using scan because the process involves applying a fixed set of values to the combinational block inputs (represented by V_1) and determining if the outputs possess the correct functional values. Stuck-at fault testing is referred to as a DC test because no timing requirements exist, i.e., delays are irrelevant. As discussed at the beginning of this section, the deep submicron era brought with it more occurrences of timing related failures and the need to apply delay tests. The 2-vector requirement for delay testing can be solved in two ways as discussed earlier. Launch-on-shift (LOS) derives V_2 by shifting the scanned in vector V_1 by 1 bit position using the scan chain. Launch-on-capture (LOC) derives V_2 from the outputs of the previous combinational block, shown as B_1 in Fig. 5 for testing paths in B_2. In both cases, it is not possible to choose V_2 arbitrarily as shown for the standard form in Fig. 4. It is important to recognize that these constraints exist (they are often ignored) and that the effectiveness of deriving delay tests for detecting HT will be negatively impacted by them.

Another issue that is often ignored deals with obtaining accurate timing information for paths. The timing diagram shown in Fig. 4 shows that it should be possible to set the **launch-capture interval (LCI)**, i.e., the interval of time between the application of V_2 at t_1 and the capture event at t_2, to any arbitrary value. Unfortunately, this is not the case. The external tester (ATE) driving the clock pin on the chip is limited in how close consecutive edges of *Clk* can be placed. Moreover, most applications of delay tests for manufacturing defects only need to determine if the chip runs at the operational clock frequency. As a consequence, the LCI is typically fixed for all tests and only upper bounds on the delays of paths within the chip can be obtained. Therefore, HT detection methods that require picosecond resolutions for individual path delays will require alternative clocking strategies and/or additional DFT components to be incorporated on the IC, which are described below.

A last important issue regarding path delay testing is related to circuit hazards. Combinational logic blocks often possess instances of *reconvergent fanout*. A simple example is shown in Fig. 6(a) for a NAND gate implementation of the XOR function. The integers inside the NAND gates represent one possible assignment of gate delays. The test sequence $AB = \{01,11\}$ is designed to test the highlighted path but in fact propagates logic transitions along both branches of the *fanout* point *C*. The timing diagram shown on the right side of Fig. 6(b) identifies a 'glitch' on the output *F* that is created by differences in the relative delays of these two paths.

Although this test is classified by the manufacturing test community as *robust*, the glitch introduces uncertainty for the security community in cases where the precise delay of the highlighted path is needed. The three transitions that occur on *F* each represent the delay of a subpath in the circuit, with the first, leftmost edge in this case corresponding to the highlighted path. Although subpath information might prove useful in providing additional HT coverage, process variations render this information challenging to leverage because of the difficulty associated with deciding

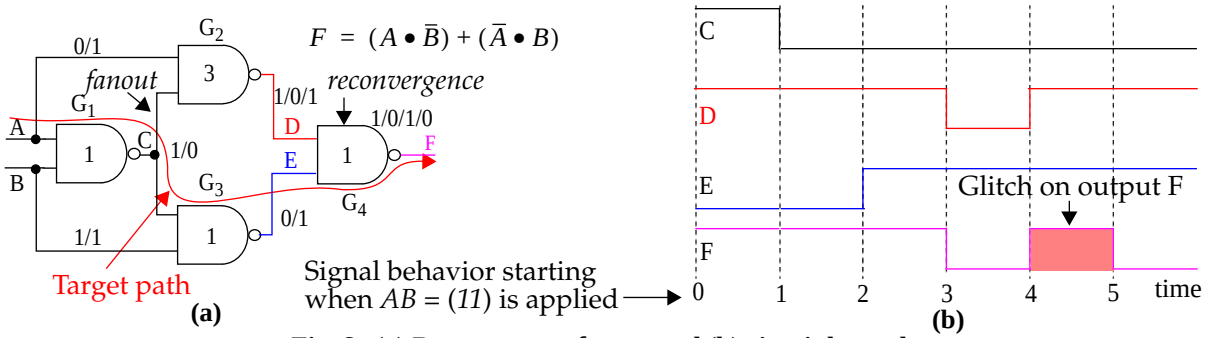


Fig. 6. (a) Reconvergent-fanout and (b) circuit hazards.

which edge corresponds to which subpath. In other words, the same test applied to a different chip with different assignments of delays to the NAND gates may reorder the edges or may in fact result in only single transition, i.e., the glitch disappears altogether. All major synthesis tools are oblivious to hazards, making them very common in synthesized implementations of functional units. Special logic synthesis algorithms are needed to construct circuits that are hazard-free, but hazard-free implementations usually have large area overheads and therefore are rarely used. Unfortunately, hazards are largely ignored in many proposed HT test generation strategies even though they can invalidate tests and raise false alarms.

4.1.2 Similarities and Distinctions of Delay Test for Manufacturing Defects and HT

Unlike logic-based testing, the goals of testing for defects and testing for HT using path delay tests are very similar. Path delay tests for defects are designed to determine if an imperfection introduced during the fabrication process causes a signal propagating along a path to emerge later than designed. Similarly, path delay tests for HT are designed to determine if an adversary has added fanout to logic gate inputs and outputs, i.e., additional wires that monitor the state of the IC (**trigger**), or inserted additional gates in series with the original design as a means of modifying its function (**payload**). Both of these scenarios also cause the delay of paths to increase.

The main distinguishing characteristic between defects and HT relates to *false positives*. False positives are situations in which a test for an HT indicates it is present when in fact it is not. This issue is less relevant for defects, and can be minimized using modern ATPG tool flows. False positives can occur for HT when the detection method does not adequately account for normal delay variations introduced by process variations. The cost associated with false positive detection decisions is very different for defects and HT. A false positive in manufacturing test results in a defect-free chip being falsely discarded, while a false positive HT detection can initiate a very expensive and time consuming reverse engineering process of the IC.

False negatives, on the other hand, need to be handled by both manufacturing defect and HT testing communities. False negatives are situations in which a defect or HT exists and it is not detected by the applied tests. False negatives can occur in either application either because the measurement technique does not provide sufficient resolution or because the applied tests do not provide adequate coverage. The cost associated with false negatives can be high in either case, resulting in system failure once the IC is installed in a customer application.

4.1.3 High Resolution Path Delay Measurement Techniques

Delay-locked loop (DLL), phase-locked loops (PLLs) and digital clock managers (DCM) are on-chip IP blocks responsible for maintaining phase alignment with external oscillators and for creating multiple internal clocks at different frequencies and with specific phase shifts[1]. They can be used to create the *Clk* signal shown in Fig. 4 for path delay testing. Although automatic test

equipment (ATE) can be used to carry out path delay testing, on-chip clock and phase shift mechanisms generally provide higher accuracy and resolution because off-chip parasitic resistor-inductor-capacitor (RLC) components are eliminated and noise sources are reduced. Many of the HT detection techniques described in subsequent sections depend on high resolution timing measurements, making on-chip techniques better suited.

Fig. 7 shows three examples of measurement techniques that can be used to provide fine-grained timing resolution. The first, called *Single-Clock scheme* (or ***clock sweeping***), requires repeated application of a 2-vector sequence (Fig. 7(a)). On each iteration, the *frequency* of C_1 is increased, which moves the launch and capture edges, i.e., the LCI, of the *Clk* signal closer together. The process is halted as soon as a condition is met or violated, which is usually related to whether the Capture FF successfully captures the functional value produced by vector V_2 (see Fig. 4). An estimate of the path delay is computed as $1/frequency_{final}$ where $frequency_{final}$ is the stop point frequency. Although this scheme requires the fewest resources, i.e., only one clock tree is included on the chip, it lower bounds the length of the path that can be measured. For example, short paths would require a very high frequency clock, which creates undesirable secondary effects, e.g., power supply noise, that make it difficult to obtain accurate timing measurements. Single-Clock schemes which use an externally-generated (ATE) clock constrain the minimum path length even further.

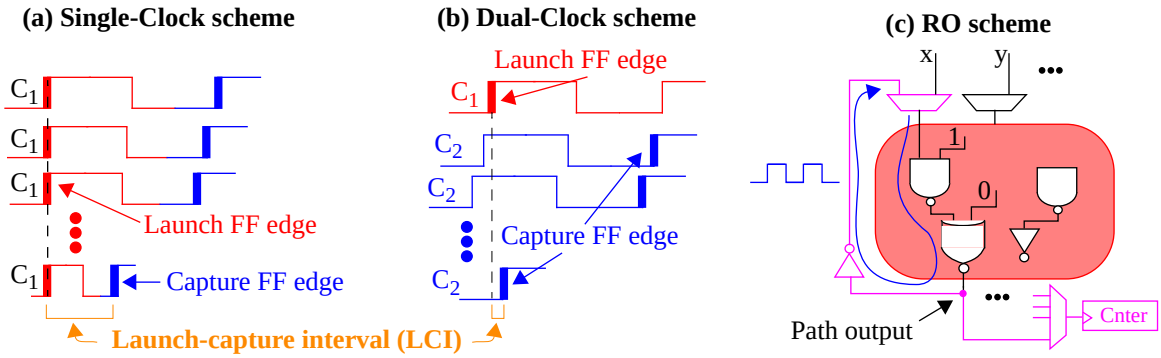


Fig. 7. Path delay measurement techniques.

The second, called *Dual-Clock* scheme (or ***clock strobing***), also requires repeated application of the 2-vector sequence [37-38]. On each iteration, the *phase* of the capture clock C_2 is decremented by a small Δt relative to C_1 as shown in Fig. 7(b). The additional overhead introduced by the second clock tree is offset by the benefit of being able to time a path of any length. This is possible because the two clock networks are independent and modern clock manager IP designs are capable of allowing the time base of C_2 to be very precisely shifted. Moreover, the power supply noise issue mentioned above is also mitigated because only two clock edges are required to carry out a launch-capture delay test instead of three.

The third timing mechanism, referred to as the *RO* scheme, is shown in Fig. 7(c) [44][1]. It adds the components shown in magenta to the design. Paths in the circuit are timed by creating a ring oscillator (RO) configuration where the output of a path is connected back to the input of the path using a MUX (and optionally a NOT gate as shown). A timing measurement is performed by

1. FPGA vendors commonly refer to IP blocks for clock control as DCMs.
1. A similar scheme called Path RO is proposed earlier by the authors of [39], but for application to design-for-manufacturability.

enabling the MUX connection and then allowing the path to 'ring' for a specific time interval. A counter (*Cnter*) is used to record the number of oscillations. This is accomplished by tieing the output signal from the path to the clock input of the counter. The actual path delay is obtained by dividing the time interval by the counter value (note, the NOT gate and MUX add two gate delays to the delay of the actual path). No launch-capture event is required in this scheme. Therefore, the clock noise associated with high frequency clocks in the *Single-Clock* scheme are eliminated. The main drawback is related to the limited number of paths that can be timed in this fashion. For example, paths that have hazards as discussed in reference to Fig. 6 produce artifacts in the count values. As discussed, hazards are very common in combinational logic circuits, and therefore, they will negatively impact HT coverage.

A fourth alternative, called a time-to-digital converter (TDC), is shown Fig. 8 [25][26][59]. Similar to the *RO* scheme, it eliminates clock strobing, and therefore, is able to obtain path delay measurements that better represent *mission mode* path delays. The TDC is an example of a 'flash' converter, which is a class of converters that digitize path delays very quickly. The *Path Select Unit* shown on the left is responsible for selecting a pair of paths, one of which can be the clock signal. The *Delay Chain Unit* shown on the right is responsible for creating a digital representation of the relative difference between the delays of the two input paths, P_{Ax} and P_{Bx}. The arrival of a rising or falling transition on one path creates the first edge in the delay chain (labeled *first* in the figure), while a transition on the second path generates the trailing edge (labeled *second*). The width of the initial (leftmost) pulse shown in red represents the delay difference between the two signals being timed. The pulse propagates along the delay chain as shown by the annotations along the top of the figure.

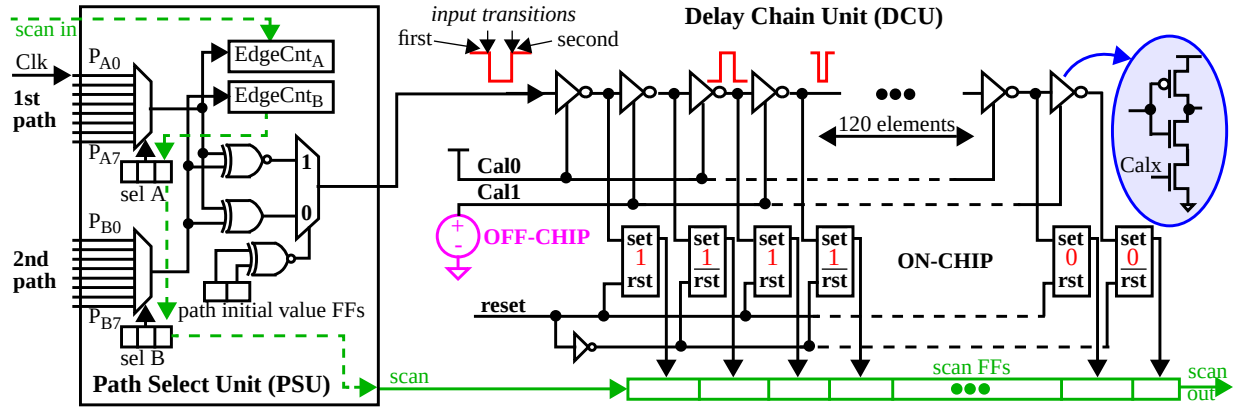


Fig. 8. Time-to-Digital Converter (TDC).

The inverters in the delay chain include an additional series-inserted *NFET* transistor as shown by the callout on the far right. An analog control signal labeled *Calx* is used to control the pull-down strength of the inverter, with higher gate voltages allowing faster operation. The inverter chain is configured with two such control signals, *Cal0* and *Cal1*. The combination of the two allows independent control over the propagation speed of the *first* and *second* edges. A calibration process is carried out in advance of making delay measurements to determine the best values of *Cal0* and *Cal1*. These analog control signals are set to allow the worst-case (widest) pulse to propagate through most of the inverters before 'disappearing'. The pulse disappears when the second edge catches up to the first edge. The calibration process is described later in Section 5.11.

The output of the inverters in the delay chain also each connect to a 'set-reset' latch. The presence of a negative pulse (for odd inverters) or positive pulse (for even inverters) changes the latch

value from 0-to-1. A digital *thermometer code* (TC), i.e., a sequences of 1's followed by zero or more 0's is produced in the sequence of latches after a test completes. The TC can be converted into a discretized delay value (if needed) using pulse width information applied during the calibration process. In addition to being very fast (less than 100 ns per measurement), the TDC is also resilient to some types of circuit hazards. For example a series of pulses can be introduced by circuit hazards but only the widest one determines the TC value (shorter pulses die out earlier in the delay chain). The *EdgeCnt* components in the *Path Select Unit* can be used to decide when hazards are present.

A fifth scheme, called REBEL in [26], also uses a delay chain to obtain timing information. REBEL is a light-weight ***embedded test structure*** that combines the delay chain component of the TDC (without the pulse shrinking characteristic) with the *clock strobing* technique referred to in Fig. 7. A significant benefit of REBEL over the TDC is complete resilience to circuit hazards. In fact, REBEL is able to provide timing information regarding each of the edges associated with hazards in a single launch-capture test. As indicated earlier, the edges produced by circuit hazards each represent the delay of some internal segment in the functional unit. Although process variations add uncertainty and diminish their usefulness, as discussed above, the ability to instantly have knowledge of their presence adds robustness to the delay measurement process and can help reduce the likelihood of false negative HT detection decisions.

4.2 Dealing with Process Variations

A significant benefit of techniques designed to detect HT in fabricated chips is the availability of a ***golden model*** of the IC. The assumption made by most of the techniques described in Section 5 is that the HT is introduced by changing the layout, via masks manipulation or through other fabrication process related steps. Therefore, all design data prior to the mask and chip fabrication steps, e.g., HDL, schematic and even the geometric layout data itself, is considered trusted. A golden model, and simulation data derived from it, provides a trusted reference to which hardware data can be compared. Path delay methods attempt to identify anomalies in the hardware data that cannot be explained by the golden model.

The most significant challenge associated with detecting HT with path delay testing is distinguishing between changes in delay introduced by a HT and those introduced by process variation effects. Failing to distinguish between these two types of delay variations leads to false positive and false negative HT detection decisions. The former declares an HT is present when it is not, while the latter fails to detect the presence of an actual HT. Minimizing false positive and false negative rates is a critical design parameter of an HT detection technique.

There are three basic approaches for dealing with process variation effects in HT methods. The first method, called **GoldenSim-based** and **GoldenChip-based**, creates simulation models or uses HT-free chips, resp. to characterize the HT-free space. The second, called **PCM-based**, uses data from *process control monitors* (PCM) to 'tune' the boundaries of the HT-free space derived from golden models using chip-measured test structure data. The third, called **Chip-Centric**, creates a *nominal* simulation model and *calibrates and averages* path delays to the nominal model (or data from HT-free chips). All approaches create a *bounded HT-free space* that represents normal variations in path delays introduced by process variations and/or measurement noise. Data collected from the test chips is compared with this bounded HT-free space. Data points that fall outside the boundaries are called **outliers**, e.g., are path delays that exceed the limits defined by the HT-free space. Chips that produce outlier data points are considered HT candidates.

The three approaches are graphically portrayed in Fig. 9 and are described in more detail throughout Section 5 as needed. The 2-D shapes labeled "Simulations with process variations

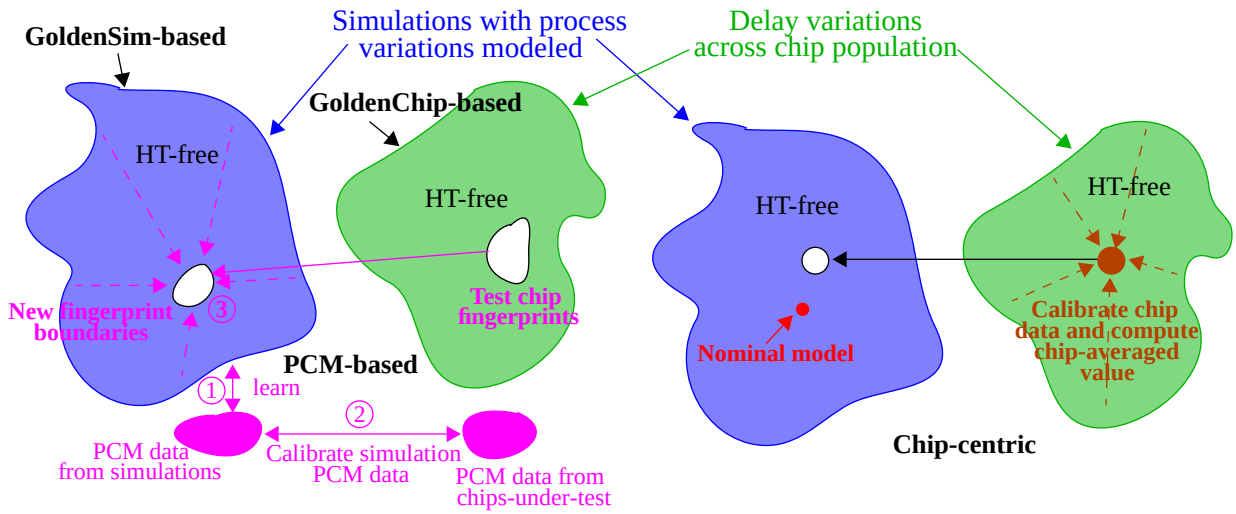


Fig. 9. Mechanisms to account for process variations using a *golden* model approach for HT detection.

modeled" and "Delay variations across chip population" can in fact be multi-dimensional, with each dimension representing one path delay or one of multiple features extracted from the set of path delays using statistical techniques such as principle component analysis (PCA).

GoldenChip-based and **GoldenSim-based** techniques train a classifier using HT-free data from chips or simulations, resp. GoldenChip-based methods measure delays from HT-free chips, which are then destructively validated to be HT-free using techniques from Section 3.1.3. GoldenSim-based methods typically use data from Spice-level simulations of a resistor-capacitor-transistor (RC-transistor) model of the *golden* design. Both techniques can be expensive in terms of reverse-engineering effort, model development and simulation time. Delayering technologies utilized for GoldenChip-based methods can take weeks or months. For GoldenSim-based methods, CAD tools such as Mentor Graphics Calibre must first be used to create the RC-transistor models of the layout using complex process models obtained from the foundry in which the chips are fabricated. The modeling files can be very large, e.g., 100's of MB, even for relatively small designs on order of 20,000 gates, and simulation times can easily extend to weeks and months when performing transient simulations with only a couple hundred input vectors. The effort required to construct and/or confirm the HT-free boundaries using either technique is very large and is often under-reported.

Of even greater concern for GoldenSim-based techniques is the level of mismatch that can exist between the simulation results and the hardware. Foundry models in advanced technologies have become very complex, providing the user with a variety of statistical evaluation methodologies including *fixed corners* and *Monte Carlo* options. Fixed corner models are provided to enable the user to predict worst-case and best-case performance of the chip by modeling the range of global process shifts that can occur over time. Unfortunately, this typically expands the HT-free space beyond what is required to represent the behavior of the chips-under-test. The expansion leads to a decrease in the sensitivity of HT methods and increases the level of mismatch between simulation and hardware data. Moreover, foundry models typically provide limited capabilities for modeling within-die variation effects, making it difficult to predict the uncertainties related to specific hardware path delays. These modeling and simulation challenges are compounded by measurement noise that occurs during chip testing, and by non-zero jitter and drift tolerances introduced by the tester during the generation and delivery of high frequency clocks. Taken together, these issues work to increase in the possibility of false positive and false negative HT detection deci-

sions.

4.3 Test Vector Generation Strategies

A last issue deals with an important distinction that exists between fault models used in manufacturing test and those required for detecting HT. The manufacturing test community developed several fault models, including **transition delay faults** and **path delay faults**, for dealing with timing problems resulting from a wide variety of defect mechanisms. For example, the transition delay fault (**TDF**) model assumes defects occur on individual *nodes* in the circuit, and that they manifest as slow-to-rise and slow-to-fall signal behaviors at those nodes. The path delay fault (**PDF**) model, on the other hand, makes no such assumptions, i.e., it accounts for defects which may in fact be distributed across one or more logic gates and wires that define the paths, and as a result, the PDF model provides more complete information about the integrity of the tested chip.

Unfortunately, obtaining 100% PDF coverage requires all (or a large fractions) of the paths in a chip to be tested. For even moderately sized circuits, the costs associated with the generation and application of a complete PDF test set is prohibitive. This is true because the number of paths can be exponentially related to the number of inputs to the chip (or functional unit). Therefore, most chip companies generate and apply TDF vectors instead because the number of such tests is linear to the number of circuit nodes in the design. Fortunately, for the security and trust community, the TDF model is a better match to the types of malicious modifications an adversary is likely to make to the layout. The node-oriented TDF model used for defects is leveraged by a large fraction of the proposed HT detection techniques described in Section 5.

There are two important points to consider with regard to test generation for HT detection. The first relates to the options that are available when the TDF model is used. Although far fewer tests are required under the TDF model to obtain high levels of HT coverage (when compared to the PDF model), there are typically many choices for the *path* that is sensitized through each of the nodes. A variety of techniques are proposed by authors of published work including random vectors, an ***incremental-coverage*** strategy driven by the sequence of vectors generated so far, traditional TDF vectors, or, in some cases, the test generation strategy is left unspecified. Others leverage the TDF model and direct ATPG to target the **shortest paths** through the node because the additional delay added by the HT (via fanout load or gate insertion) has a larger fractional impact on the path delay. The traditional TDF model for defects, on the other hand, typically target the **longest paths** as a mechanism to ensure that at least this subset of tested paths meet the timing constraints.

The length of the path relates to the second important point regarding test generation. Automatic test equipment is outfitted for manufacturing test, which is focused on testing the longest paths. For test cost reasons, it is common that only one clock frequency is used to apply TDF tests to the chip because the primary goal of manufacturing test is to ensure that the delays of all tested paths are less than the upper bound defined by the clock period. The most sensitive tests for defects therefore are those that test the longest paths. This is true because the longest paths minimize the **slack**, i.e., the difference between the clock period and the delay of the tested path.

Many believe that these manufacturing test constraints for defects are not sufficient for providing high levels of HT coverage. This is reflected in the proposed use of *clock sweeping*, *clock strobing* and other on-chip embedded test structures for obtaining precise measurements of path delays. In other words, the slacks inherent in tests for defects provide too many opportunities for adversaries to ‘hide’ the additional delay of the HT in the slack, and therefore, a paradigm shift is required regarding the manner in which delay testing is carried out on the test floor. Clock sweeping and clock strobing are expensive in terms of test time, and HT detection techniques which use

these clocking strategies need to account for the higher levels of clock noise associated with high frequency clocks and invalidations introduced by circuit hazards. It remains to be seen how the economic tradeoffs of delay-based HT detection schemes will play out.

5. HT Detection Methods Based on Path Delay Analysis

This section is dedicated to describing a selected subset of the proposed HT detection strategies that have been proposed over the last decade. Our goal is to describe methods that offer some unique perspective, and therefore, this exposition does not provide an exhaustive survey of every published paper on the topic. The choice to include a description of a published work was based on whether it promoted the state-of-the-art in at least one of the three technical domains described earlier, including the path measurement technique, the statistical method used to distinguish between HT anomalies and process variation effects, and test vector generation strategy. The techniques are presented chronologically instead of by technical domain. The latter organization presented challenges because many techniques propose solutions to more than one domain.

5.1 Early HT Detection Techniques and On-chip Measurement Methods

The first works on using path delays for HT detection are described in [40] and [41]. The primary focus of each paper is on only one of the technical domains, in particular [40] on a statistical method for distinguishing between process variations effects and HT, and [41] on a high resolution on-chip measurement technique.

In [40], the HT detection method assumes high resolution path delay measurements are available, i.e., no measurement strategy is proposed. Although not explicitly stated, the test vector generation strategy appears to be based on the **standard transition delay fault** model. They base their detection method on the **GoldenChip-based** model described in Section 4.2. A multivariate statistical technique is used to extract distinguishing features from the full set of path delays. HT-free chips are used to construct the HT-free boundaries, which they refer to as a *fingerprint*. The fingerprints define the boundaries of the shape labeled “Delay variations across chip population” shown on the left side of Fig. 9. HT are detected by comparing the delay fingerprints measured from the untrusted test chips with the boundaries defined by the HT-free fingerprints.

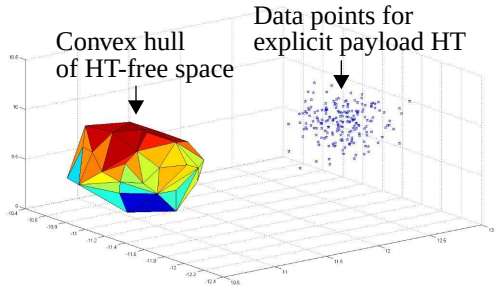
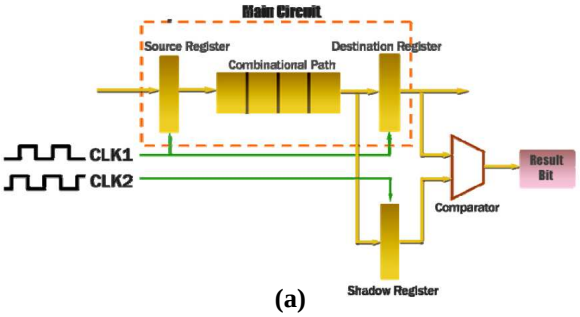


Fig. 10. Convex hull characterization showing detection of an HT [40].

They demonstrate their technique using simulations in which ATPG-derived 2-vector sequences are applied to a DES functional unit. Principle component analysis (PCA) is used to extract distinguishing features from a set of 10,432 simulated path delays as a means of reducing the HT-free space to a 3-D structure. A statistical technique based on a *convex hull* characterization of the HT-space is used to define the boundaries for each of the 64 outputs of DES. Four HT are inserted into another set of models, with three representing *explicit payload HT* and one representing an *implicit payload HT.* The explicit payload HT inserts one or more additional gates in series with paths in the HT-free design, while the implicit payload HT is represented as a simple counter with no ability to change the functional characteristics of DES. They show the explicit

payload HT are easily detected (see Fig. 10) while the implicit payload HT is only detected approx. 36% of the time.

A high resolution on-chip path delay measurement technique is proposed in [41], which is extended in [42] to include a **GoldenSim-based** HT detection strategy. The measurement technique is based on the **Dual-Clock** scheme described in reference to Fig. 7. A set of *shadow registers* are added to each of the outputs from the combinational components of the design, next to the capture FFs or *Destination Register*s as shown in Fig. 11(a). The second clock of the Dual-Clock scheme, *CLK2*, is used to drive the clock inputs of the shadow registers. *CLK2* is generated as a 'fine-phase-shift' adjusted version of *CLK1* using a DCM on the FPGA.



(a)

$$t_{\text{path}} = t_{CLK1} - n_p \Delta t_p$$

(b)

Fig. 11. (a) On-chip path timing architecture proposed in [41] and (b) equation giving actual path delay.

The process of measuring the path delay of the Combination Path from Fig. 11 begins by setting the phase shift of *CLK2* to a small negative value, on order of 10 to 100 ps (see Fig. 7). A 2-vector sequence is applied to the Source Registers using a launch-capture test. The *Comparator*, also added to the design, is used to determine if the captured values in the Destination and Shadow register are the same or different. If they are the same, which is the case when the clock strobe operation begins, the negative phase shift difference between *CLK1* and *CLK2* is increased and the same 2-vector sequence is applied. This process is repeated until the comparator indicates the values are different. The actual delay of the path is computed by multiplying the number of phase shifts, n_p, by the Δt_p provided by each phase shift increment. Subtracting this value from the *CLK1* period yields an estimate of the path delay, t_{path}, as given by the equation in Fig. 11(b).

The extended work in [42] investigates the detection capabilities of a **GoldenSim-based** technique on a 8-bit Braum Multiplier functional unit. HT are modeled as series-inserted 2-inverter chains. The proposed method derives a path delay distribution using simulation data, but is constrained by the measurement resolution provided by the timing technique. Process variations are modeled by varying transistor threshold voltage, V_{th}, and transistor channel length, L_{eff}, in simulations with and without HT. Data from these simulations is used to define the boundaries of the shape labeled "Simulations with process variations modeled" shown on the left side of Fig. 9. The amount of skew in the mean of the distributions is used as the detection criteria. The results using four inserted HT show that three can be detected and the last one is detectable on some outputs but not others.

5.2 Ring Oscillator-based HT Detection Approaches

A distributed set of ring oscillators (RO) is proposed in [43] as a means of detecting HT. The array of ROs is distributed uniformly across the (x,y) space of the functional unit as shown in Fig. 12. The detection criteria is based on HT power consumption. If a HT is present, the test stimulus applied to the functional unit may cause at least some gates within the HT to switch (referred to as ***partial activation*** in [1]), and the HT will necessarily consume power. The additional HT power

consumption creates localized voltage drops on the supply rail (V_{DD}) that can be detected by comparing the delay of a nearby RO with that of a HT-free chip or simulation. The delay variations introduced by the HT-switching-induced voltage drops in the RO are integrated by the RO over time and are reflected in a counter value. A counter is connected to the RO using MUX2 as shown along the bottom of Fig. 12. MUX1 is used to select and enable one RO at a time as a random, LFSR-based test vector sequence is applied to the inputs of the functional unit. This process is repeated for the *n* ROs (12 in figure) with the set of counts representing the *signature* for the chip.

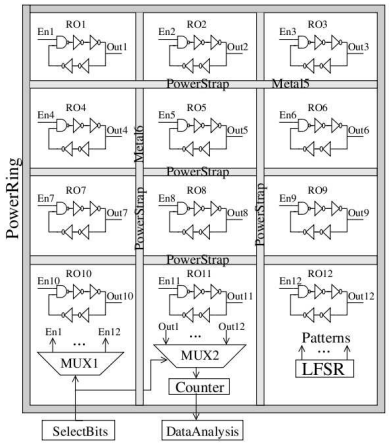


Fig. 12. RO distribution architecture proposed in [43] for detecting HT switching activity.

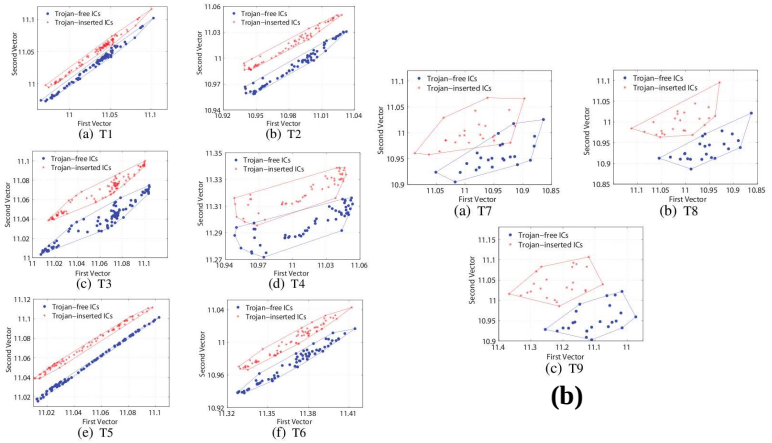


Fig. 13. Subset of results from [43] using proposed *advanced outlier analysis* method, (a) simulation results and (b) FPGA results.

Process parameter variations are accounted for by collecting data from a large number of HT-free chips (**GoldenChip-based** model) and a statistical analysis is applied to the signature using principle component analysis (PCA) and correlation analysis. One of the techniques, called *advanced outlier analysis*, analyzes data obtained from pairings of ROs as a means of detecting **regional** power droop anomalies. The results shown in Fig. 13 plot the RO pairings that show the best detection capability for each of the 6 HT inserted into the design. The points within each graph represent simulation results with process variations modeled in (a) and results derived from FPGA experiments in (b). The separation of the red HT-inserted and blue HT-free points illustrates that all 6 HT are detected in every simulation process model in (a) and nearly all are detected in every FPGA in (b).

The authors of [44] propose a *Design-for-Trust* (DFTr) technique designed to detect HT by creating ROs from the functional paths in a design[1]. They propose an algorithm that selects paths with the maximum number of ‘unsecured gates’, i.e., gates that have not already been included in other ROs, i.e., a method characterized as an ***incremental-coverage*** driven strategy. For each selected path, a MUX, a control signal and optionally an inverter are added to the design to complete the ring. Automatic test pattern generation (ATPG) is then used to generate input patterns that place ***non-dominant*** values on the ***off-path*** inputs of gates sensitized by the ring. Non-dominant refers to gate input values that do not determine the gate’s output value by themselves, e.g., a ‘1’ is the non-dominant value for an AND gate. Off-path inputs refer to side inputs of gates in the

1. A related design-for-manufacturability scheme was proposed earlier in [39] for measuring critical path delays.

RO that are not on the sensitized path of the RO. These conditions ensure the RO will 'ring' when enabled by the control signal. Fig. 14 shows the ISCAS-85 benchmark circuit *C17* configured with a set of ROs.

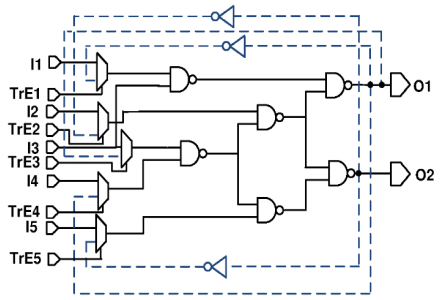


Fig. 14. ISCAS-85 benchmark C17 configured with a set of ROs using the DFTr method proposed in [44].

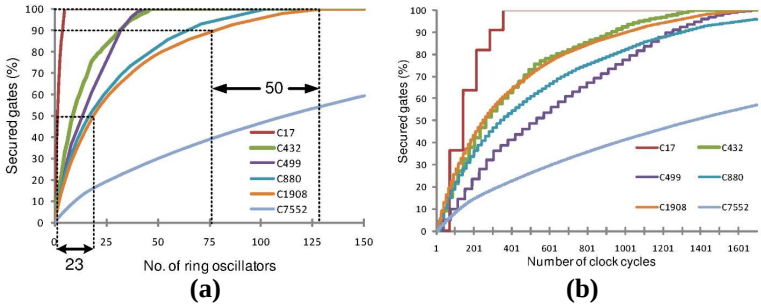


Fig. 15. (a) Number of ROs required as a function of coverage for six ISCAS-85 benchmark circuits, and (b) test time required to obtain $F_{measured}$ for each chip [44].

Simulation experiments with process variation effects modeled are carried out to determine the golden frequencies, F_{golden} (**GoldenSim-based** model). HT detection is carried out by comparing $F_{measured}$ obtained from each of the untrusted test chips with F_{golden}. The authors implemented their technique on a Xilinx Spartan 3 FPGA using six ISCAS-85 benchmark circuits. The number of ROs required to attain a specific level of HT coverage is given in Fig. 15(a). As is typical of test generation for manufacturing defects, coverage per RO drops dramatically for coverage targets above 90%. Test times are given in Fig. 15(b) which shows similar trends. Although the proposed technique is very promising, the authors do not address hazards that occur within circuits with reconvergent-fanout.

5.3 Lightweight On-Chip Path Timing Techniques for HT Detection

Two on-chip delay measurement techniques, called TDC and REBEL, are proposed in [26]. An HT detection method which leverages REBEL [45] is described in the following. A second HT detection method which uses the TDC [59] is covered in Section 5.11.

REBEL is designed as an embedded test structure (ETS). ETS are instruments that are integrated directly into functional units as a mechanism to obtain regional, high precision information about its operational characteristics. ETS must be designed to be minimally invasive and low in overhead to avoid violating power, area and performance constraints associated with the functional unit. REBEL satisfies these ETS attributes by leveraging components in the existing scan chain architecture.

REBEL is designed to provide regional, high resolution measurements of path delays. It also addresses the clock noise issue discussed in Section 4.1 related to using Single-Clock schemes to obtain timing information for short paths. The architecture of REBEL allows paths within the functional unit to be extended along a delay chain, effectively eliminating the need for high frequency clocks. The delay chain is created using the existing capture FFs attached to the outputs of the functional unit. Fig. 16(a) shows an example configuration with REBEL integrated into a pipelined functional unit. A path-under-test (PUT) within the functional unit is highlighted as well as the delay chain that is created through the capture FFs. Row Control Logic is added to the design to enable one of the path outputs to be selected as the target of the timing measurement process.

A path is timed by applying a 2-vector sequence to the inputs of the functional unit. The transition along the PUT emerges at the output and propagates along the delay chain. The capture

edge of the clock creates a digital snapshot of the transition by storing in the Capture FFs a sequence of digital values which represent its behavior over time. Each of the snapshots immediately reveal whether the propagating signal has more than one transition, i.e., whether a hazard is present or not.

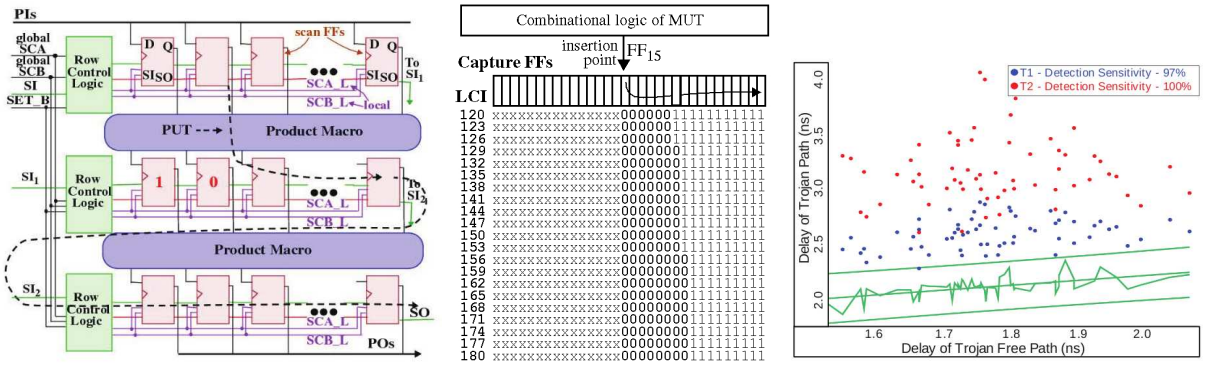


Fig. 16. (a) Integration of an embedded test structure called REBEL in a pipelined functional unit as described in [45], (b) sequence of digital 'snapshots' stored in the REBEL delay chain with successive rows showing an increasing launch-capture interval (LCI) and (c) illustration of regression analysis applied to path delays measured from 62 chips for an HT-free path (x-axis) and a second path (y-axis) with and without the inclusion of HT.

A sequence of digital snapshots are shown in Fig. 16(b) as rows labeled 120 to 180, which represent a *clock sweeping* sequence of LCIs as described earlier in reference to the Single-Clock scheme of Fig. 7. For each successive LCI, the propagating falling transition driving the input of the capture FF_{15} is given more time to propagate along the delay chain. The path tested in this example does not generate any type of hazard (is hazard-free), otherwise one or more of the snapshots would show interleaved '1's in the sequence of '0's. In practice, the LCI test sequence is actually applied in reverse, starting with 180, because larger LCI increase the amount of temporal information stored in the Capture FFs regarding the propagating transition. The larger time window provides a better opportunity to detect hazards which can invalidate the HT test.

As proof-of-concept, a 90 nm chip is designed and tested which allows paths in the functional unit (in this case, an 8-function floating point unit or FPU) to be reconfigured with and without a HT. Although the experimental results presented use hardware data to define the HT-free space (**GoldenChip-based** model), the authors also acknowledge a **GoldenSim-based** approach is possible.

The statistical method proposed in [45] is based on linear regression analysis. Fig. 16 shows a subset of the results in which delays from one HT-free path (x-axis) and a second path that can be configured with and without HT (y-axis) are compared. The HT-free space is highlighted in green and is delineated by three sigma regression bands. It is derived using data from 62 copies of the test chips configured without the HT included in the second path. The red and blue points each represent the results with the second path configured with one of two possible HT. All but one of the data points fall outside the regression limits, and are therefore classified as detected. Additional results for a larger set of HT are reported in the paper.

5.4 Self-Authentication: A Golden Model-Free HT Detection Method

A golden-model-free HT detection method is described in [46] which inserts a framework of *HT detection sensors* into the layout representation of a design[1]. The sensors are designed as replicas of common sequences of logic gates (path-sequences) that already exist in the design. Cus-

tom CAD tools are used to decompose the timing graph of a design to identify a set of commonly occurring *delay features*. Delay features correspond to layout specific patterns of gates and interconnect that share common geometries and sensitivities to process parameter variations. Similarity among features is determined by evaluating the *changes* in delay that occur when the path-sequences are subjected to similar process conditions. Two sequences are considered *similar* if the changes in their delay track within a small error tolerance.

Once a set of target path-sequences are identified in the design, a set of matching sensors are integrated into the layout in close proximity to the targeted path-sequences. After fabrication, the delay fingerprints of the sensors and corresponding full-length paths (that contain the path-sequence(s)) are measured. Data from each of the sensors is used to construct an HT-free delay range, which captures the measurement noise profile for the sensor. A similar process is carried out for each of the paths. The delay associated with other components of the full-length paths, in which the path-sequences are contained, is accounted for using variation-aware expressions. A nominal simulation or static timing analysis estimate is used to determine the nominal delay of the sensor, which, in combination with the measured delays, allows the delay of the full-length paths to be predicted. Correlation analysis is used to compare the predicted and measured delay ranges for the sensors and paths. Outliers are considered anomalies introduced by HT in either the sensor, the path or both. Fig. 17 provides a flow diagram of the self-authentication process and shows examples of the HT insertion and detection scenarios considered [46].

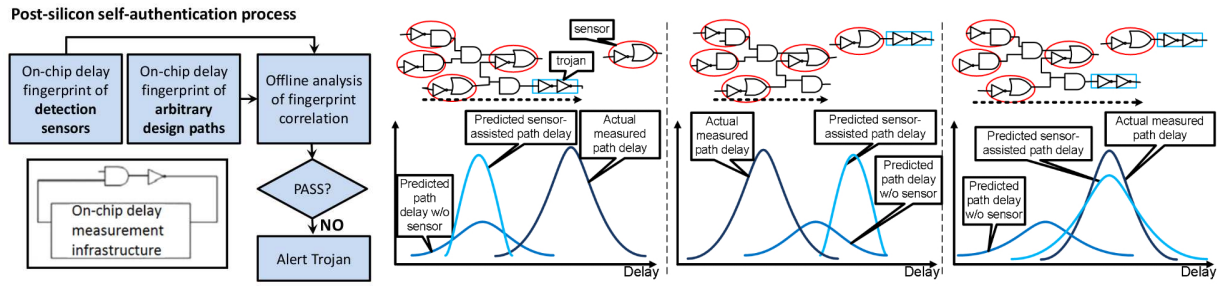


Fig. 17. (a) Self-authentication chip testing process proposed in [46] and (b) sensor (both simulated and measured) and measured full-length path delay range illustrations under three HT attack models. Mismatches in the overlap among the sensor distributions, or low levels of correlation between sensor and path distributions is flagged as an HT detection.

The authors assume on- or off-chip delay measurement schemes such as those described in reference to Fig. 7 are available. The sensors act as *silicon-anchor* points for calibration, and therefore the proposed HT detection technique shares similarities to the *process control monitor* approach described in Section 4.2 and referred to as **PCM-based**.

The authors apply the technique to the ISCAS-89 benchmark circuits, synthesized to layouts using a 90 nm TSMC technology. Process variations are modeled in the simulations by varying major device parameters within 10% of nominal using a Monte Carlo selection process. A multi-level hierarchical model of the layout is processed as a means of partitioning the design into regions where it is assumed that process variations are more highly correlated. Sensors are identified and designed but constrained to use no more the 15% additional area in the layout. A set of paths are randomly selected to serve as the HT-insertion points for evaluation of the method. A set of 30 HT are inserted into each path with varying amounts of delay to determine sensitivity, and

1. A self-referencing technique is also proposed earlier in [47] but is based on the correlation of transient power supply currents produced by replicas of the functional unit.

10K process models are created and simulated (300K per path). HT detection rates are shown to improve from between 2% to 16% when compared to a similar method that does not leverage sensors as a sensitivity-enhancing technique.

5.5 Linear Programming Methods and Test Point Insertion for HT Detection

A linear programming method is proposed to derive leakage, power and delay characteristics of individual gates based on solving a system of equations, referred to as *gate-level characterization* (GLC) [48-50]. Chip measurements of power and delay are used in the system of equations, along with estimates of measurement errors, to derive scaling factors for the parameters associated with the logic gates in the design.

GLC is combined with a test point insertion technique in [49] as a means of improving coverage of HT. The authors propose to add FFs (test points) to components of the design that exhibit *reconvergent-fanout*. An example of reconvergent-fanout is shown in Fig. 18 from [49] where both the fanout and reconvergence points are identified. The task of generating path delay tests for circuits which contain complex reconvergent-fanout networks is an NP-complete problem. Automatic test pattern generation (ATPG) algorithms can fail to determine 2-vector sequences that are able to test the individual paths within reconvergent-fanout blocks, such as those labeled ‘Path 1’ and ‘Path 2’, even when such tests exist. Although in the example it is trivial to derive test patterns that test these two paths individually (node assignments are shown that allow Path 1 to be tested by itself), there are other more complex configurations which require an exhaustive search, proportional to 2^n, to find suitable 2-vector sequences.

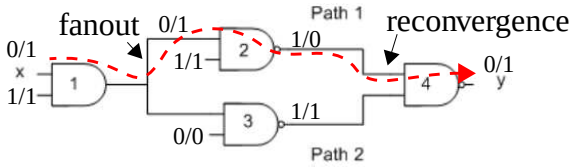


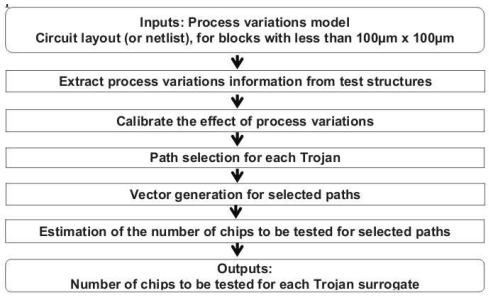
Fig. 18. (a) Example of reconvergent-fanout [49].

The authors propose an algorithm that first identifies all paths that can be easily tested and then a set of paths in reconvergent-fanout logic structures that are the best candidates for test point insertion. A SAT-based process is proposed to select input vectors that maximize the number of independent linear equations for application of GLC. A second circuit partitioning scheme based on *maximum fanout free cones* is proposed in [50] to increase the number of delay access points within the circuit as a means of improving coverage further for large designs.

5.6 Process Calibration and Test Vector Selection for Enhancing HT Detection

A delay calibration technique is described in [51] that leverages information obtained from test structures as a means of detecting HT delay anomalies that are very small, i.e., within the margin of those introduced by process parameter variation effects in advanced technologies. The test structure measurements are used to estimate the global mean shift in delay introduced by variations in the process parameters for each chip. Based on the estimate, the mean value for the paths in the region of the embedded test structures are calibrated to eliminate the mean shift.

The process flow proposed in [51] is shown in Fig. 19. The first step is to extract information from the embedded test structures, such as ring oscillators, as a means of obtaining process parameter information for each chip. Test structures are added to the layout in regions close to the functional unit to be tested. This ensures that both global process variations and systematic within-die variations are accurately captured in the measurements. Path selection and vector generation are carried out such that test cost is minimized. ATPG is constrained to generate robust



(a)

TABLE V. DELAYS MEASURED ON VARIOUS INVERTER CHAINS (*L*: INVERTER CHAIN LENGTH, σ: STD. DEV.. OF DELAY, Δ: EXTRA DELAY INDUCED BY A TROJAN, μ: MEAN DELAY, AND *N*: THE NUMBER OF CHIPS TO BE TESTED)

Case I				Case II				Case III			
L	σ/μ	Δ/μ	*N*	*L*	σ/μ	Δ/μ	*N*	*L*	σ/μ	Δ/μ	*N*
2	0.269	0.207	14	2	0.246	0.207	13	2	0.244	0.206	11
4	0.185	0.163	14	4	0.164	0.162	13	4	0.162	0.161	11
6	0.143	0.126	14	6	0.125	0.125	13	6	0.122	0.125	11
8	0.124	0.101	17	8	0.105	0.100	13	8	0.103	0.099	12
10	0.113	0.088	20	10	0.092	0.087	13	10	0.088	0.086	12
12	0.107	0.076	20	12	0.084	0.075	14	12	0.081	0.074	12

(b)

Fig. 19. (a) Process flow model proposed in [51], (b) simulation-derived delay statistics obtained by applying the proposed method under different types (global and within-die) process variations.

tests (when possible) for critical (longest) paths passing through each possible HT site, and therefore test generation is based on the **traditional TDF** model. Path delays are measured using the **Dual-Clock** scheme shown in Fig. 7 as a means of minimizing clock noise and obtaining high resolution measurements. The integration of silicon-anchor points for calibration of process variations classifies the proposed technique as **PCM-based**.

An estimate of the mean shift in each region of the chip is computed using test structure data and a *minimum mean square (MMSE) estimator.* The MMSE finds the mean $\hat{\mu}$ that minimizes the sum of the squared differences between the test structure data and the computed mean as given by Eq. 1. This estimator can then be used to calibrate all the path delays for a given chip by simply subtracting $\hat{\mu}$ from each measurement. A novel non-parametric hypothesis testing method based on a *likelihood-ratio test* is proposed which leverages integer linear programming (ILP) for determining the number of chips that need to be tested to achieve a specific confidence level against false positive and false negative HT detection decisions.

$$\sum_{i=1}^{M} (d_i - \hat{\mu})^2 \qquad \textbf{Eq. 1.}$$

The technique is evaluated on a set of inverter chains of length 2 through 12 with and without HT insertions. HT are modeled using a minimum size inverter connected to the output of the first inverter in the chain. Monte-Carlo simulations were performed using circuit models with different types of global and within-die process variations modeled, referred to as Case I (global only), Case II (across-chip random and systematic within-die only) and Case III (local random and systematic within-die only). Fig. 19(b) shows that the best results are obtained for Case III which uses simulation models *with only local process variations included*. The proposed calibration method in this case makes this possible by eliminating Case I and Case II via the test structure measurements, which minimizes the σ/μ statistical variation parameters as well as the number of required chips.

This technique is extended in [52] to address the best paths to target for HT detection. In contrast to their earlier work, the authors argue that the shortest paths through each HT site maximize detection sensitivity (see **shortest path TDF** discussion in Section 4.3). The column labeled Δ/μ in Fig. 19(b) expresses the impact of the HT on path delay, and is the focus of the current work. Given that the adversary's goal is to minimize the impact of the HT on path delay, shorter paths are better suited to reveal these small delay variations because the (constant) delay added by the HT becomes a larger fraction of the total delay for short paths. A similar argument regarding the effect of process variations also holds. In particular, the σ of variations is approx. proportional to

the nominal delay of the path, i.e., shorter paths have smaller σ. This characteristic is illustrated in Fig. 20 which shows the path distributions for a long path (top) and short path (bottom) with and without HT. The HT, represented as an 'additional fanout', creates a more distinguishable shift in the short path distribution when normalized as a fraction of the total width of the distribution. In both cases, the HT adds only 8 ps to the path delay but the smaller σ corresponding to the shorter path provides a higher level of confidence in detecting the anomaly.

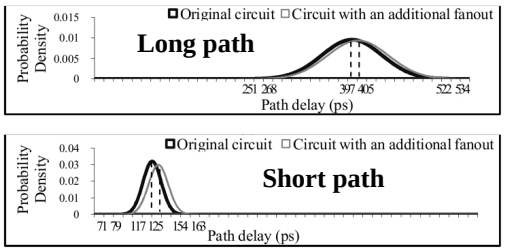


Fig. 20. Impact on path delay distribution for a long and short path, with short path showing larger fractional change [52].

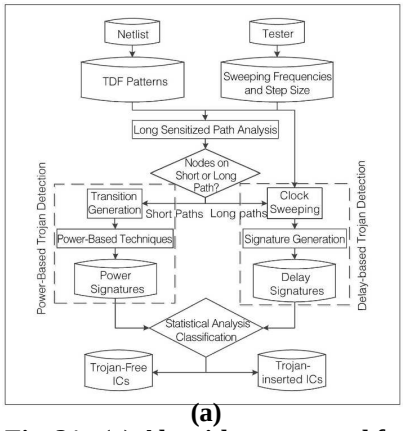
The authors also argue that shorter paths are more likely to be the targets of an HT insertion because longer paths, particularly critical paths, increase the chance of accidental discovery. Moreover, generating vectors for shorter paths is generally 'easier' for ATPG tools to accomplish because fewer side inputs need to be 'justified' (forced to specific values) in order to sensitize the path from PI to PO. The main benefit of short paths however, according to the authors, is the *reduction in the number of chips that need to be tested* (see column labeled *N* in Fig 19(b)). On the downside, shorter paths are harder to time, especially when using the Single-Clock scheme from Fig. 7, because the chip needs to be tested at much-faster-than-at-speed to obtain precise delay measurements. The Dual-Clock scheme provides a solution but it also requires the addition of a second clock tree as described in [41].

An algorithm is presented that both selects the shortest path through each circuit node (each HT site) and enforces constraints on the *robustness* of the test to ensure the target path is in fact the path tested by the 2-vector sequence. The authors present simulation results using the ISCAS-89 benchmark circuits that show a 2.1X improvement in test cost over a traditional TDF strategy. They further show that the improvement increases to 4.51X when combined with the calibration technique proposed in [51].

5.7 Clock Sweeping for HT Detection

The authors of [53] propose a *clock sweeping* method to address sensitivity issues associated with using a traditional TDF model and the path delay fault (PDF) model for detecting HT. Clock sweeping refers to the **Single-Clock** scheme referenced earlier in Fig. 7 in which the clock frequency is incrementally increased (by a fixed *step size*) and a 2-vector sequence is applied repeatedly until a *delay fault* is detected in the Capture FFs for one or more of the tested paths.

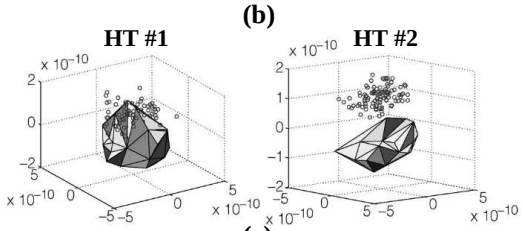
The authors propose to generate tests using the TDF model described earlier and acknowledge that short paths whose delay is smaller than the maximum frequency are not testable because of the limits of ATE and clock noise. The algorithm that they propose is shown in Fig. 21(a). It partitions TDF tests into two groups, those sensitizing long paths to be tested in the proposed HT delay technique and those sensitizing short paths to be tested using a power-based HT method. They argue that long paths experience less switching activity because more conditions need to be met in order to sensitize them. Therefore, power-based methods are less effective for detecting HT on these paths.



(a)

	P1/FF1	P1/FF2	...	P2/FF1	P2/FF3	...	PP_1/FFm
IC 1	f_6	f_{14}	...	f_3	f_{20}	...	f_{10}
IC 2	f_7	f_{12}	...	f_4	f_{21}	...	f_9
IC 3	f_5	f_{11}	...	f_3	f_{19}	...	f_{10}
...	...	...	...	...	...	...	...
IC n	f_8	f_{14}	...	f_4	f_{23}	...	f_{12}

(b)



(c)

Fig. 21. (a) Algorithm proposed for HT detection in [53], (b) chip signatures recording the first failing frequency for pattern (Px) and Capture FF (FFx), and (c) MDS/convex hull results for 2 HT.

The failing frequencies for long paths are recorded in a table as shown in Fig. 21(b). Chips are listed on rows while the columns identify the pattern, *Px*, and Capture FF, FF_x, of the tested paths. A multidimensional scaling (MDS) statistical method is proposed for distinguishing between delay variations introduced by process variation effects and those introduced by HT. MDS leverages PCA to map from a higher dimensional space to a smaller space. Unlike the technique proposed in [40] however, they configure MDS to preserve signature components that represent dissimilarities introduced by HT delay anomalies in the lower dimensional space. A 3-D convex hull is constructed using signatures from HT-free chips and outlier data points from the untrusted chips are classified as HT candidates. Their detection technique therefore is based on the **Golden-Sim-based** or **GoldenChip-based** model described in reference to Fig. 9.

An ISCAS-89 benchmark circuit s38417 is used in their simulation experiments to validate the technique. Simulation models representing process variations are constructed by varying threshold voltage, oxide thickness and channel length over 5% of nominal both globally and locally to model within-die variations. A total of six HT are introduced in a layout representation of the benchmark circuit with varying trigger and payload configurations. The clock frequency range and step size used for clock sweeping is set to 700 MHz to 1.5 GHz and 10 ps, resp. The results of applying MDS and constructing a convex hull are shown in Fig. 21(c). The detection rate for HT #1 is 64% while the rate for HT #2 (and the remaining four HT not shown) is 100%. A similar set of results are obtained in hardware experiments using a set of 44 FPGAs.

5.8 A Golden Chip-Free Method for HT Detection

The authors of [54] propose the use of *process control monitors* (PCMs) that are designed to eliminate the need for a set of HT-free *golden chips*. PCMs are in-line test structures traditionally inserted by process engineers for tracking wafer-level variations in transistor parameters such as threshold voltage. The authors use the delay of a special path as a surrogate for a PCM as a silicon calibration method. Path delays from this PCM are measured from the test chips and used to improve the accuracy of the classification boundary first obtained from simulation data. This detection strategy is therefore **PCM-based** as discussed in Section 4.2.

The authors employ *non-linear regression* and *kernel mean matching* techniques to learn the relationship between PCM data and side-channel fingerprints, in this case, output power measurements from a set of 40 wireless cryptographic chips which instantiate AES with and without HT. A series of 'learned' boundaries are incrementally tuned as each of five statistical transformations are applied using simulation data obtained from the PCM and AES process models, and from the

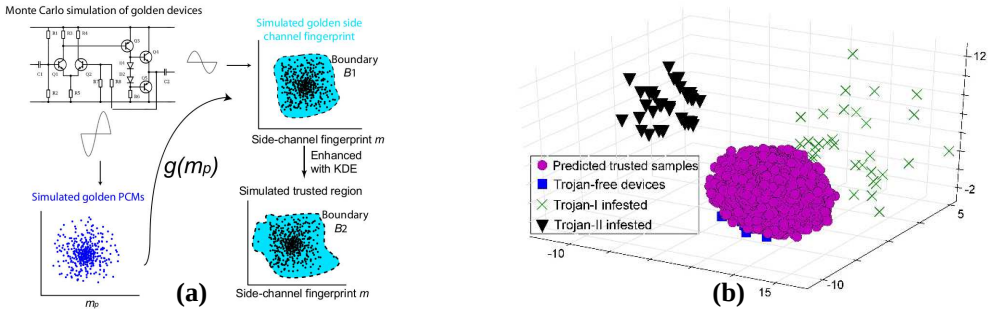


Fig. 22. (a) Initial simulation-based statistical transformations designed to iteratively learn the best boundaries associated with the HT-free fingerprint space for a wireless cryptographic IC from [54], (b) top three principle components from PCA analysis after application of proposed statistical learning process. All 80 HT are detected and all but three of the HT-free chips (of 40) are classified correctly.

PCM and power measurements from the test chips. Fig. 22(a) shows the first set of transformations which are derived from simulation data, illustrating transformations that take place in the shape and boundaries of the HT-free space. The remaining transformations are derived using PCM and path delay data measured from a set of HT-free chips and are illustrated in their paper. Fig. 22(b) shows experimental results in which all 80 HT are correctly classified as HT-infested while only 3 of the HT-free chips are classified incorrectly, i.e., are false positives.

5.9 HT Detection by Comparing Paths with Structural Symmetry

An HT detection method based on validating delay consistency among instances of distinct transistor-level paths with the same topology is proposed in [55]. Symmetry is defined by considering both the structural characteristics of the logic gate(s) and state assignments on its inputs under each of the vectors of an applied 2-vector sequence. For example, a NAND gate exhibits symmetry in delay by having two identical pull-up paths through its two PMOS transistors and when input transitions are crafted to exercise each of these paths, at a time, during a delay test. An HT detection algorithm is proposed that first identifies transistor-level symmetry in the netlist or layout and then adds constraints to ATPG algorithms to test pairs of pairs that exhibit this symmetry. A ***self-referencing*** detection algorithm is proposed that compares the delays of symmetrical paths and classifies a chip as having an HT when the two path delays are not identical within a threshold.

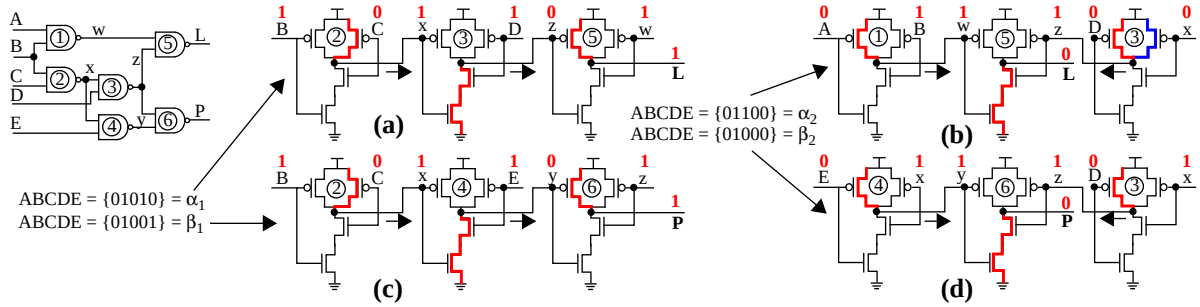


Fig. 23. Transistor-level symmetry illustration adapted from [55].

The authors present an example of transistor-level symmetry using the ISCAS-85 c17 benchmark circuit, which is reproduced with enhancements in Fig. 23. The gate-level netlist of c17 is shown on the far left while transistor-level netlists representing subsets of the netlist are shown in (a) through (d). The transistor-level diagrams are annotated with numbers to enable the NAND gates to be cross-referenced to the c17 schematic. The transistor level schematics along the top and bottom rows represent the two paths that exhibit symmetry. The first 2-vector sequence of the

symmetry pair is labeled α_1 and α_2 while the second 2-vector sequence is labeled β_1 and β_2. The 2-vector sequences used as the tests are given as (α_1,α_2) and (β_1,β_2).

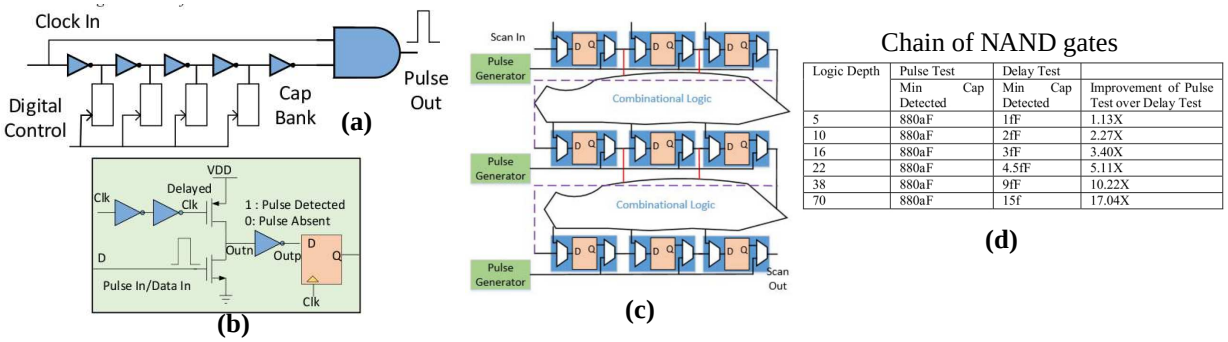
The red highlighted components show the pull-up and pull-down paths that connect the output of each NAND gate to one of the supply rails, which is determined by the logic state imposed by each of the four vectors. For example, the outputs of the 3 NAND gates in (a) are connected to V_{DD}, GND and V_{DD} for gates labeled 2, 3 and 5. The key observation is the consistency of the highlighting between (a)-(c) and (b)-(d) and the fact that the actual gates in these pairs are different except for one of the gates. In other words, the application of the two 2-vector sequences test the same pull-up and pull-down paths in the NAND gates but do so along different paths in c17. Given the NAND gates have identical layout structures, the path delays are expected to be nearly identical. Therefore, if an adversary inserts one or more payload gates in series with either of these paths, the delays will be different and can be flagged as a malicious modification. Simulation and FPGA results are shown to demonstrate this concept.

The delay changes introduced by global shifts in process variations (and within-die variations to some degree) are eliminated because the comparisons are made between paths on the same chip, and preferably in close proximity. Therefore, none of the *golden model* techniques referenced in Section 4.2 for dealing with process variations are required. However, margins are needed to account for measurement noise and routing differences in the two paths, otherwise the false positive rates will be high. The authors indicate that finding structural symmetries in the layout and then deriving qualifying test patterns can be challenging given the large number of constraints that must be satisfied to ensure consistency in the behaviors of the pull-up and pull-down components of the tested paths. This *feature* can be argued is a benefit because it makes the task difficult for the adversary to carry out and then defeat the technique by inserting the HT such that the delays of symmetrical pairs of paths remain consistent.

5.10 HT Detection using Pulse Propagation

A high-resolution HT detection method is proposed in [56] that is based on propagating pulses along digital logic paths. HT detection is accomplished by detecting whether pulses survive, i.e., do not die out, before reaching the Capture FF where they are detected. Minimum pulse widths that allow the gates along the path to sustain the pulse are constrained by only one of the gates along the path, in particular, the gate that has the largest rise + fall time. The authors argue that this characteristic greatly enhances the HT detection sensitivity of their method to capacitive loading effects over other delay testing methods, particularly for long paths and when considering process variation effects. Delay variations introduced by process variation effects are cumulative and therefore, the HT-free boundaries or *margins* associated with standard delay methods must be increased for longer paths, which reduce their sensitivity to small, fixed-sized variations in delay introduced by HT. On the other hand, pulses will shrink when they encounter the capacitive load of an HT, and will die out at the gate that was used to determine the minimum pulse width for the path (note: this assumes the HT insertion occurs before the gate that was used to define the minimum-sized pulse). Therefore, the authors argue that HT detection sensitivity remains constant and is independent of the length of the path. The embedded components needed for pulse generation and pulse detection can be designed as shown in Fig. 24(a) and (b), resp., and these components can be shared among multiple FFs, as shown for the pipelined architecture in Fig. 24(c).

An algorithm is presented that uses *n* random test patterns that are each evaluated through simulation using *k* pulses of different widths. Test for paths that are able to propagate the pulse from a Launch FF and to a Capture FF are deemed valid. The authors refer to these paths as *sin-*



Chain of NAND gates

Logic Depth	Pulse Test Min Cap Detected	Delay Test Min Cap Detected	Improvement of Pulse Test over Delay Test
5	880aF	1fF	1.13X
10	880aF	2fF	2.27X
16	880aF	3fF	3.40X
22	880aF	4.5fF	5.11X
38	880aF	9fF	10.22X
70	880aF	15f	17.04X

(d)

Fig. 24. (a) Proposed pulse generator, (b) pulse detector, (c) an example illustrating sharing within a pipelined architecture, and (d) simulation results comparing proposed technique (labeled Pulse Test) with standard delay test (labeled Delay Test) [56].

gle-path sensitizable[1]). Simulations are again used to determine the minimum pulse width for each path using worst-case process models. HT are emulated on each node of every path using different capacitive loads to determine the minimum capacitance that succeeds in 'killing' the propagating pulse.

The proposed method is validated using simulation experiments on a chain of NAND gates, a ripple carry adder and 4x4 multiplier. Process variations are modeled by changing threshold voltage (V_t) by +/- 10% globally and +/- 10% locally. The detection results for the chain of NAND gates are shown in Fig. 24(d), with 'Pulse Test' results corresponding to the proposed technique and 'Delay Test' identifying results using a standard delay test strategy. The columns labeled 'Min Cap Detected' represent the smallest HT capacitive load that was detectable for the paths of different lengths specified by the rows. The last column expresses the improvement in sensitivity of the proposed method over the standard delay test method, and supports the claim that the pulse method remains sensitive to small HT even when the path length becomes very large. Similar results are obtained for the other functional units as reported in [56].

5.11 Chip-Centric Calibration Techniques for HT Detection

The authors of [57][59] propose HT detection methods which use actual path delay measurements, in contrast to PCMs and other types of on-chip test structures, as a mechanism to calibrate for global shifts and within-die process variation effects[2]. These methods represent variants of the **Chip-Centric** technique described in Section 4.2. Chip-centric techniques can potentially provide higher levels of sensitivity to HT because the path delays used in the detection method also serve as the basis for calibration. Moreover, by using chip-measured path delays to shrink the HT-free space, as depicted on the right side of Fig. 9, such methods can also simplify the development of a simulation-based *golden model*, as demonstrated in [59].

The authors of [57] **average** path delays measured from a set of chips to reduce the adverse impact of both inter-chip and intra-die process variation effects on HT detection sensitivity. The proposed *golden model* is based on hardware measurements of delays from HT-free chips, i.e., design and simulation data are not used to develop the HT-free space. The data collected from the chips is multi-dimensional. The authors use # to symbolize the chip number, *P* to represent the

1. Single-path sensitizable refers to paths that are hazard-free robust testable, indicating all side-inputs along the path must remain constant under both applied vectors.
2. Note, the path delay technique described in [59] is based on the same concept presented earlier in [58] which uses leakage currents.

pattern (2-vector sequence) number, N_P to represent the number of patterns, α to identify functional unit outputs (Capture FFs) and N_α to represent the number of outputs.

Calibration of inter-chip (global) process variation effects on path delays uses a *centering* operation in which the delays D under all patterns P to an output α for chip # are averaged and subtracted from each of the raw delays as given by Eq. 2. Therefore, the method uses the distribution of delays to each output α for calibration of the global *mean* shift in path delays that occurs within chip #. A second *centering* operation is then performed to further reduce intra-die variations which averages the globally-calibrated delays to each output α across all chip outputs as given by Eq. 3. This chip-wide average is then subtracted from the raw path delays for a chip to provide a set of locally-calibrated delays.

$$\mathfrak{D}_P(\alpha,\#) = \mathfrak{D}(P,\alpha,\#) - \frac{\sum_P \mathfrak{D}(P,\alpha,\#)}{N_P} \quad \textbf{Eq. 2.}$$

$$\mathfrak{D}_{P,\alpha}(\#) = \mathfrak{D}_P(\alpha,\#) - \frac{\sum_\alpha \mathfrak{D}_P(\alpha,\#)}{N_\alpha} \quad \textbf{Eq. 3.}$$

$$RP_{P,\alpha,\beta}(\#) = \frac{\mathfrak{D}_{P,\alpha}(\#)}{\mathfrak{D}_{P,\beta}(\#)} \quad \textbf{Eq. 4.}$$

$$Dg_{P,\alpha,\beta}^{\#_{test}} = \frac{RP_{P,\alpha,\beta}(\#_{test}) - \overline{RP}_{P,\alpha,\beta}(\#_{GM})}{\sigma_{P,\alpha,\beta}} \quad \textbf{Eq. 5.}$$

The ratios of two locally-calibrated delays for a pattern P are used in the formulation of a *golden model*, each referred to as a ***relative performance*** metric, $RP_{P,\alpha,\beta}$ as given by Eq. 4. A matrix of relative performances is constructed for each chip # and the mean value $\overline{RP}_{P,\alpha,\beta}$ computed across all HT-free chips, N_{GM}, is used as the references for comparison of the $RP_{P,\alpha,\beta}$ values computed from the untrusted chips. A margin referred to as the *coefficient of irrelevance* is proposed for dealing with false positive HT detections. It is defined as the standard deviation, $\sigma_{P,\alpha,\beta}$, of the $RP_{P,\alpha,\beta}$ computed using N_{GM} HT-free chips. The threshold that bounds the HT-free space is given by Eq. 5, and is referred to as a ***distinguisher***.

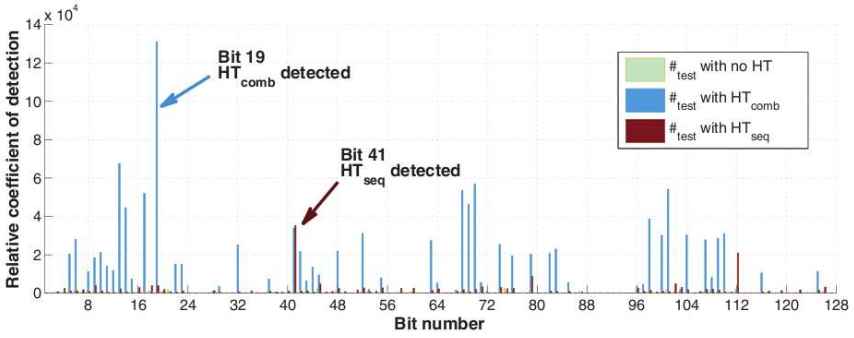


Fig. 25. HT detection results as presented in [57].

The technique is validated using a set of four Xilinx Spartan FPGAs programmed with a AES-128 functional unit and modified in a second design to include one combinational and one sequential HT. The golden model is built using delays measured from the AES-128 without the HT. The **Single-Clock** scheme (or *clock sweeping* from Section 4.1.3) is used with a step size of 35 ps and a frequency range from 100 MHz to 121.2 MHz. Test vector selection is performed randomly, i.e.,

no test vector generation strategy is proposed. A set of 50 patterns (plaintexts) are used as the test vector set and paths from all 128 bits of the AES are monitored. Path delays shorter than 8.25 ns (1/121.2 MHz) are ignored. The authors report on a subset of the *distinguishers*, in particular, the distinguishers which produced the maximum value for each of the 128 outputs when computed using the HT-free data and data from the two HT experiments. The results are shown in Fig. 25, with highlights indicating the outputs that provide the highest levels of confidence in detecting the two HT.

Although the technique proposed in [57] is demonstrated to work well, the averaging techniques that the authors employ do not deal directly with intra-chip process variations. The technique presented in [59] (discussed below), on the other hand, averages delays across chips for each path and 2-vector sequence, instead of across vectors and outputs. Within-die variations have been shown to have a significant random component in each chip instance [60] and therefore, a path-by-path averaging strategy is likely to be more effective in reducing unwanted intra-chip variations. Moreover, the strategy proposed in [57] only calibrates for the global shift in the *mean* values of path delays introduced by inter-chip process variation effects. The technique described in the following also considers *scaling* effects.

A **chip-averaging** HT detection method that calibrates for both intra-chip and inter-chip process variations and measures path delays using an on-chip time-to-digital converter (TDC) is proposed in [59]. The TDC was described earlier in reference to Fig. 8 in Section 4.1.3. The TDC provides approx. 25 ps of timing resolution, is very fast, e.g., no clock strobing or clock sweeping operation is required, and can be multiplexed and shared across a large number of the functional unit outputs. The method is also classified as **Chip-Centric** but unlike [57], does not depend on a set of golden chips. Rather, a *golden simulation model* is used to characterize the HT-free space. The development of the golden model requires only a **single nominal simulation** to be run for each of the applied 2-vector sequences, and therefore the approach significantly reduces the level of effort and time required over previously proposed simulation-based golden model approaches. This is possible because the calibration processes are geared toward deriving a nominal chip-averaged-delay (**CAD**) value for each path from hardware data, and therefore, process variation effects do not need to be accounted for in the golden model.

Calibration and chip-averaging are designed to reduce performance differences and the adverse effects of process variations on delay while preserving any type of systematic variation that shows up in all (or a large subset) of the tested chips. Chip-averaging leverages a key difference between random process variations and HT anomalies; random variations average to 0 while HT anomalies introduce systematic differences that survive the averaging process.

The authors validate the method using data collected from 44 copies of an ASIC fabricated in a 90 nm technology which has two exact copies of the layout of an AES functional unit, one representing the original design and one with five embedded HT. A layout of the chip showing the two copies of the AES and four instances of the TDC is shown in Fig. 26(a). The two 8-to-1 multiplexers shown in the block diagram of the TDC from Fig. 8 connect to 15 of the 128 outputs of the AES (the 16th input is connected to the *Clk*). The two copies of the TDC in each AES instance allow signals propagating to 30 of the outputs of AES to be timed against the *Clk*.

The calibration process used to reduce inter-chip process variations is carried out in advance of the HT detection procedure on each chip separately. Similar to the HT detection process, calibration involves measuring delays from paths of various lengths within the functional unit on each chip. Unlike HT detection, the goal of calibration is to tune the control signals, *Cal0* and *Cal1*, of the TDC as a means of *shifting* (and *scaling*) the delay distribution obtained for each chip to a

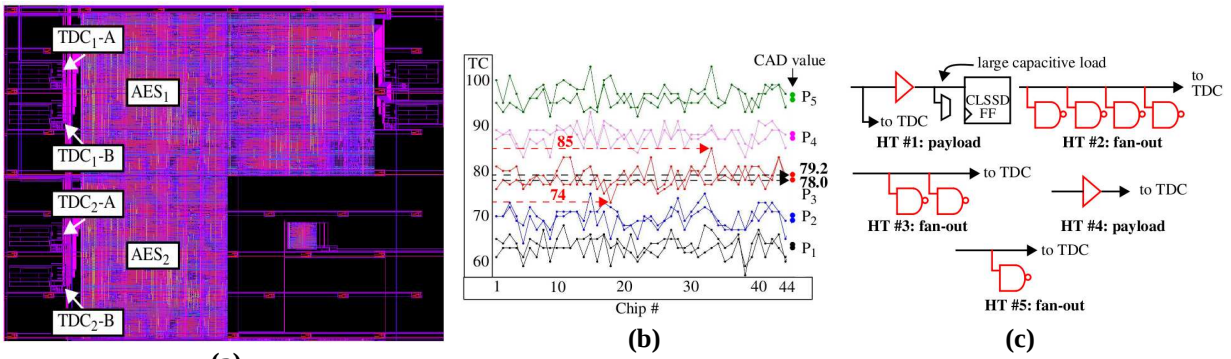


Fig. 26. (a) Chip layout showing two copies of the AES [26], (b) TCs from 44 chips with CAD values shown on far right, and (c) configurations of HT added to AES_2.

fixed mean value. From Section 4.1.3, the output of the TDC is a thermometer code (TC), i.e., an integer value between 0 and 120, that represents the relative delay difference between the *Clk* and the path being tested. The fixed mean value is set to the halfway point (60). By using the same fixed mean value for all chips, this process effectively standardizes the TCs, thereby eliminating most of the delay variations introduced by chip-to-chip process variation effects.

The chip-averaging technique is designed to remove the remaining intra-chip variation that exists in the path delays. Once the TDC is calibrated, a set of TDF-based vectors designed to test each possible HT site in a hazard-free fashion is applied to the chips. The HT detection method is applied once data from all or a large sample, e.g., 50 or more, chips is collected. A **chip-averaged-delay** (CAD) value is computed for each tested path by averaging the TC delays obtained from all chips. The CAD averages and ideally eliminates random within-die variations, making it possible to observe very small systematic differences which occur in the chip values but are not present in a Spice-level simulation of the *nominal* model.

As an illustration, Fig. 26(b) plots the raw TC values for 5 HT-free paths of different lengths. The x-axis lists the chip, 1 to 44, for each TC value on the y-axis. The two curves represent the data collected from each of the two nearly identical AES instantiations shown in Fig. 26(a). The variations in the data points across chips and between the AES instantiations is what remains after calibration, and is attributed to intra-die variations and measurement noise. The CAD values for each of the 5 paths are shown as the last point of the waveforms on the far right. The chip-averaging effect is reflected in the 'closeness' of the 2 points computed from the 44 chips of each AES instantiation. The layout of the two AES instantiations are identical and therefore, ideally, the CAD values should be superimposed. Although this is not the case, the CAD values are closer than most of raw TC values for any given chip. This reduces the boundaries associated with the HT-free space, which in turn, improves the HT detection sensitivity of the proposed method.

The authors validate the detection sensitivity of the method by measuring the delay anomalies introduced by five layout-inserted HT. Fig. 26(c) gives schematic-level diagrams illustrating the structure and insertion points of the HT, which are highlighted in red. Four fanout HT and one series-inserted HT are added to the layout of AES_2 by replacing filler cells and connecting the inputs and outputs of the HT as shown by the schematic. A *nominal* simulation model of the AES layout and TDC are created using Mentor Graphics Calibre XRC extractor and the foundry-provided models for the 90 nm technology in which the chips were fabricated. Transient simulations using Cadence Spectra are carried out to obtain the TC values associated with the nominal model.

The graphs shown in Fig. 27 plot two sets of results, (a) plots the simulation nominal model

data against the HT-infested AES_2 data while (b) plots the simulation data against the HT-free AES_1 data for the same 20 paths. The y-axis plots a **DCAD** value, which is simply the difference between the simulation TC value and hardware-derived CAD values. HT that introduce larger anomalies therefore generate larger DCAD values. The paths are sorted left-to-right according to the magnitude of the HT delay anomaly, with the largest DCAD values on the left. The red curves represent data collected from paths that include one of the HT shown in Fig. 26(c) while the black curves represent data from HT-free paths. The displacement of the red curves upwards with respect to the black curves in (a) portrays the presence of the delay anomaly introduced by the HT. The curves in (b), on the other hand, show the DCAD values for these same paths from AES_1 (which does not include the HT) are interleaved with the HT-free (black) curves.

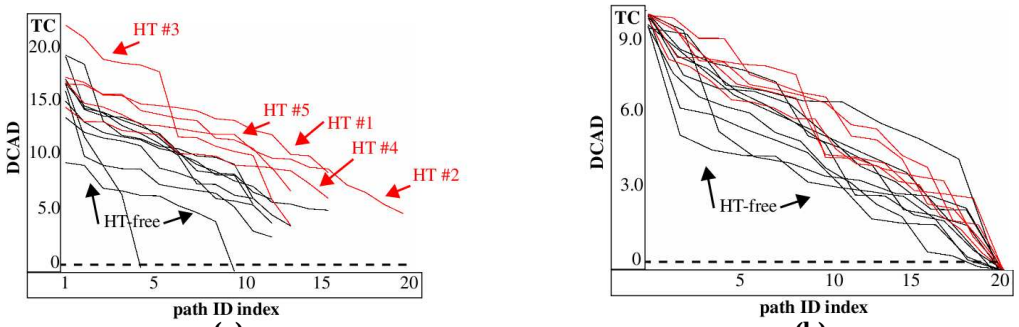
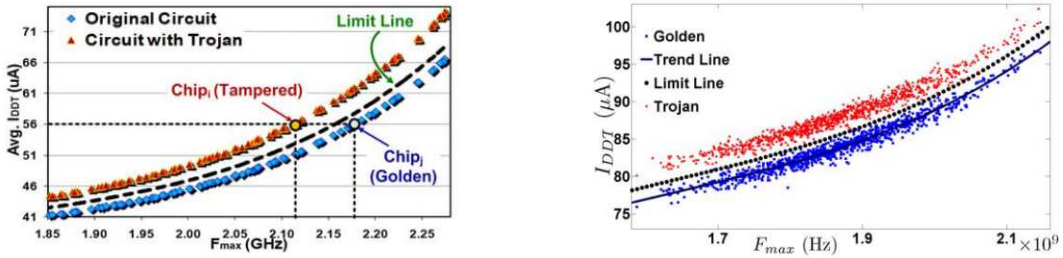


Fig. 27. (a) DCAD values of golden simulation model against HT-infested AES_2 and (b) DCAD value of golden simulation model against HT-free AES_1 from [59].

6. Multi-Parameter Detection Methods

The authors of [61] leverage correlations between maximum operating frequency, F_{max}, and transient current, I_{DDT}, as a mechanism to enhance the HT detection sensitivity of I_{DDT}. Multiple-parameter side-channel analysis refers to the joint analysis of two or more circuit parameters, such as power and delay, as a means of accounting for process variation effects or to provide higher levels of confidence that an HT exists through corroborative evidence, or a lack thereof, from multiple signal sources. This concept is portrayed in Fig. 28(a) which plots I_{DDT} against F_{max}. Here, simulation experiments are used to show an embedded HT effects I_{DDT} because of additional HT switching activity but does not impact F_{max}. The mismatch in the correlation of I_{DDT} and F_{max} allows the HT to be identified in the "Tampered" I_{DDT} curve that would otherwise not be possible. F_{max} is effectively used to track process variation effects. The distinction is blurred to some degree with the addition of random within-die variations as shown in Fig. 28(b), but the correlation and benefit provided by F_{max} remains apparent in the displacement and separation of the red (HT) and blue (HT-free) data points. The authors note that any path or set of paths can be used for the correlation analysis to make it nearly impossible for the adversary to defeat the technique.

The authors propose a test vector generation strategy that first partitions the multi-module design into non-overlapping functional blocks as a mechanism to amplify the HT I_{DDT} contribution (signal) over normal background I_{DDT} (noise). Vectors optimized to target HT nodes are selected and directed at testing one of the blocks while simultaneously minimizing activity in other functional blocks. The test vectors for I_{DDT} and a separate set for F_{max} are used in the proposed test flow to optimize correlations as shown in Fig. 28. Simulation and FPGA results are presented which validate their approach.



(a) **(b)**

Fig. 28. (a) Correlations between I_{DDT} vs. F_{max} distinguish HT-free and HT-infested chips in the presence of process noise, (b) similar analysis with random intra-die variations added [61].

7. Conclusion

Hardware Trojans (HT) represent a serious threat and a significant challenge. Side-channel techniques, such as power and delay analysis, can be argued are the most sensitive and cost-effective strategies for detecting HT. This chapter surveyed a wide variety of delay-based approaches that have been proposed over the last decade. Important technical aspects and distinctions that characterize the proposed HT detection methods can be summarized as follows:

- Path delay measurement strategy for obtaining precise measurements of path delays:
  - Clock sweeping implemented by adjusting the frequency of applied clock
  - Two clock approaches which tune the phase between launch and capture clocks (clock strobing)
  - On-chip, embedded test structures which create (tunable) delay chains
- Test stimulus strategies for HT detection:
  - Random vectors
  - Vectors generated using the traditional transition fault delay (TDF) model
  - Vectors generated from a pseudo-TDF model targeting shortest sensitizable paths
  - Pulse-stimulus-based techniques
- Approaches to account for process variation effects, both chip-to-chip and within-die:
  - HT-free space created from process simulation models
  - HT-free space created from data collected from golden (HT-free) chips (which are validated using destructive delayering techniques)
  - Simulation-derived HT-free space calibrated with hardware data from process control monitors (PCMs), ring oscillators (ROs), critical paths, etc.
  - Techniques which average path delays measured from (untrusted) chips, and compared against (nominal) simulation models or golden HT-free chips
  - Techniques which correlate multiple side-channel signals
- Design-for-trust additions, modifications and analyses to support HT detection methods:
  - Techniques which create ROs from functional unit paths
  - Techniques which add a distributed set of ROs designed to detect HT switching activity
  - Methods designed to find structural symmetry in path delays for comparison
  - Techniques which add symmetrical components to enable calibration using chip data
- Statistical HT Detection Methods:
  - Simple thresholding and linear regression-based methods
  - Advanced statistical analysis techniques which employ non-linear regression, kernel mean matching, principle component analysis, multidimensional scaling and convex hull construction
  - Ad hoc statistical techniques which leverage path delay differences, ratios and other mathematical transformations

Taken collectivity, three critical features emerge as requirements for a fully specified and effective HT detection method.

- First, traditional manufacturing test methods are not capable of providing precise measurements of path delays, which is a requirement of nearly all proposed HT detection methods. Therefore, a paradigm shift is required in the way path delay testing is carried out by automatic test equipment and/or in the capabilities of design-for-testability support structures included on the chip. Several low-cost embedded test structures were described that support high resolution on-chip measurements of path delays.
- Second, both within-die and chip-to-chip process variations pose significant limits on HT detection sensitivities and must be dealt with in a cost-effective manner. Golden model-based methods must be based on realistic assumptions regarding the availability of golden chips, and the amount of simulation time and effort required to define the boundaries of a multi-dimensional HT-free space. Golden-model-free methods must have validation techniques to guard against subversion by the adversary.
- Third, a low-cost test vector generation strategy must be developed that is effective at detecting subtle HT loading effects, and which also provides high levels of HT coverage while minimizing test cost.

Achieving all of these goals is very challenging, but the commercial acceptance of path delay testing as a mainstream HT detection strategy critically depends on low cost solutions to all three of these technical domains.

8. References

[1] X. Wang, M. Tehranipoor and J. Plusquellic, "Detecting Malicious Inclusions in Secure Hardware: Challenges and Solutions", *International Workshop on Hardware-Oriented Security and Trust*, 2008, pp. 15-19.

[2] R. S. Chakraborty, S. Narasimhan and S. Bhunia, "Hardware Trojan: Threats and Emerging Solutions", *International High Level Design Validation and Test Workshop*, 2009, pp. 166-171.

[3] M. Tehranipoor and F. Koushanfar, "A Survey of Hardware Trojan Taxonomy and Detection", *Design & Test of Computers*, Vol. 27, Issue 1, 2010, pp. 10-25.

[4] "Trustworthy Hardware: Identifying and Classifying Hardware Trojans", *Computer*, Vol. 43, Issue 10, 2010, pp. 39-46.

[5] M. Beaumont, B. Hopkins and T. Newby, "Hardware Trojans -- Prevention, Detection, Countermeasures", *Department of Defense*, Australian Government, 2011.

[6] S. Bhunia, M. Abramovici, D. Agrawal, P. Bradley, M. S. Hsiao, J. Plusquellic and M. Tehranipoor, "Protection Against Hardware Trojan Attacks: Towards a Comprehensive Solution", *Design & Test*, Vol. 30, Issue 3, 2013, pp. 6-17.

[7] N. Jacob, D. Merli, J. Heyszl, and G. Sigl, "Hardware Trojans: Current Challenges and Approaches", *IET Computers and Digital Techniques*, Vol. 8, No. 6, 2014, pp. 264-273.

[8] S. Bhunia, M. Hsiao, M. Banga, and S. Narasimhan, "Hardware Trojan Attacks: Threat Analysis and Countermeasures", *Proceedings of the IEEE*, Vol. 102, No. 8, 2014, pp. 1229-1247.

[9] https://users.ece.cmu.edu/~koopman/des_s99/sw_testing/#reference

[10] F. Wolff, C. Papachristou, S. Bhunia and R. S. Chakraborty, "Towards Trojan-Free Trusted ICs: Problem Analysis and Detection Scheme", *Design, Automation and Test in Europe*, 2008.

[11] E. Love, Y. Jin, and Y. Makris, "Proof-Carrying Hardware Intellectual Property: A Pathway to Trusted Module Acquisition", *Trans. Information Forensics Security*, Vol. 7, No. 1, 2012, pp. 25-40.

[12] M. Banga, M. Chandrasekar, L. Fang, and M. Hsiao, "Guided Test Generation for Isolation and Detection of Embedded Trojans in ICs", *Great Lakes Symposium on VLSI*, 2008, pp. 363-366.

[13] M. Banga and M. Hsiao, "A Region Based Approach for the Detection of Hardware Trojans",

Workshop on Hardware-Oriented Security and Trust, 2008, pp. 40-47.
[14] R. S. Chakraborty, F. Wolff, S. Paul, C. Papachristou, and S. Bhunia, "MERO: A Statistical Approach for Hardware Trojan Detection", *Workshop on Crytographic Hardware and Embedded Systems*, 2009, pp. 396-410.
[15] M. Banga and M. Hsiao, "A Novel Sustained Vector Technique for the Detection of Hardware Trojans", *International Conference on VLSI Design*, 2009, pp. 327-332.
[16] R. S. Chakraborty, S. Narasimhan, and S. Bhunia, "Hardware Trojan: Threats and Emerging Solutions", *International High Level Design Validation Test Workshop*, 2009, pp. 166-171.
[17] H. Salmani, M. Tehranipoor and J. Plusquellic, "A Layout-Aware Approach for Improving Localized Switching to Detect Hardware Trojans in Integrated Circuits, *International Workshop on Information Forensics and Security*, 2010.
[18] H. Salmani, M. Tehranipoor, and J. Plusquellic, "A Novel Technique for Improving Hardware Trojan Detection and Reducing Trojan Activation Time, *Trans. on VLSI Systems*, Vol. 20, No. 1, 2012, pp. 112-125.
[19] D. Karaklajic, J.-M. Schmidt, I. Verbauwhede, "Hardware Designer's Guide to Fault Attacks", *Transactions on VLSI Systems*, Vol. 21, Issue 12, 2013, pp. 2295-2306.
[20] P. Kocher, J. Jaffe and B. Jun, "Differential Power Analysis", *Advances in Cryptology*, 1999.
[21] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, B. Sunar, "Trojan Detection using IC Fingerprinting", *Symposium on Security and Privacy*, 2007, pp. 296-310.
[22] R. Rad, J. Plusquellic, M. Tehranipoor, "Sensitivity Analysis to Hardware Trojans using Power Supply Transient Signals", *Workshop on Hardware-Oriented Security and Trust*, 2008, pp. 3-7.
[23] J. Aarestad, D. Acharyya, R. Rad and J. Plusquellic, "Detecting Trojans Though Leakage Current Analysis Using Multiple Supply Pad I_{DDQ}s", *Transactions on Information Forensics and Security*, Vol. 5, Issue 4, 2010, pp. 893-904.
[24] M. Bushnell, V. D. Agrawal, "Essentials of Electronic Testing for Digital, Memory, and Mixed-Signal VLSI Circuits", Vol. 17, *Springer*, 2000.
[25] J. Kalisz, "Review of Methods for Time Interval Measurements with Picosecond Resolution", Metrologia, Vol. 41, No 1, 2003.
[26] C. Lamech, J. Aarestad, J. Plusquellic, R.M. Rad, K. Agarwal, "REBEL and TDC: Embedded Test Structures for Regional Delay Measurements", *International Conference on Computer-Aided Design*, 2011, pp. 170-177.
[27] http://techinsights.com/
[28] https://sstp.org/companies/analytical-solutions-inc
[29] J. Soden, R. Anderson and C. Henderson, "Failure Analysis Tools and Techniques -- Magic, Mystery, and Science", *International Test Conference*, Lecture Series II "Practical Aspects of IC Diagnosis and Failure Analysis: A Walk through the Process", 1996, pp. 1-11.
[30] S. R. Nassif, "Design for Variability in DSM Technologies", *International Symposium on Quality Electronic Design*, 2000.
[31] J.-J. Liou, K.-T. Cheng and D. A. Mukherjee, "Path Selection for Delay Testing of Deep Sub-Micron Devices using Statistical Performance Sensitivity Analysis", *VLSI Test Symposium*, 2000.
[32] A. K. Majhi and V. D. Agrawal, "Delay Fault Models and Coverage", *International Conference on VLSI Design*, 1998.
[33] Y. K. Malaiya and R. Narayanaswamy, "Modeling and testing for Timing Faults in Synchronous Sequential Circuits", *Design and Test of Computers*, 1(4), 1984, pp. 62-74.
[34] J. L. Carter, V. S. Iyengar and B. K. Rosen, "Efficient Test Coverage Determination for Delay Faults", *International Test Conference*, 1987, pp. 418-427.
[35] G. L. Smith, "Model for Delay Faults based upon Paths", *International Test Conference*, 1985, pp. 342-349.
[36] C. J. Lin and S. M. Reddy, "On Delay Fault Testing in Logic Circuits", *Transactions on Computer-Aided Design*, Vol. CAD-6, No. 5, 1987, pp. 694-703.
[37] D. Ernst, S. Das, S. Lee, D. Blaauw, T. Austin, T. Mudge, N. S. Kim and K. Flautneret, "Ra-

zor: Circuit-Level Correction of Timing Errors for Low-Power Operation", *Micro*, Vol. 24, No. 6, Nov. 2004, pp. 10-20.
[38] J. Li and J. Lach, "Negative-Skewed Shadow Registers for At-Speed Delay Variation Characterization", *International Conference on Computer Design*, 2007, pp. 354-359.
[39] X. Wang, M. Tehranipoor, R. Datta, "Path-RO: A Novel On-Chip Critical Path Delay Measurement under Process Variations", *International Conference on Computer-Aided Design*, 2008.
[40] Y. Jin and Y. Makris, "Hardware Trojan Detection using Path Delay Fingerprint", *Workshop on Hardware-Oriented Security and Trust*, 2008, pp. 51-57.
[41] J. Li and J. Lach, "At-Speed Delay Characterization for IC Authentication and Trojan Horse Detection", *Workshop on Hardware-Oriented Security and Trust*, 2008, pp. 8-14.
[42] D. Rai and J. Lach, "Performance of Delay-Based Trojan Detection Techniques under Parameter Variations", *International Workshop Hardware-Oriented Security and Trust*, 2009, pp. 58-65.
[43] X. Zhang, M. Tehranipoor, "RON: An On-Chip Ring Oscillator Network for Hardware Trojan Detection", *Design and Test in Europe*, 2011.
[44] J. Rajendran, V. Jyothi, O. Sinanoglu and R. Karri, "Design and Analysis of Ring Oscillator based Design-For-Trust Technique", *VLSI Test Symposium*, 2011, pp. 105-110.
[45] C. Lamech and J. Plusquellic, "Trojan Detection based on Delay Variations Measured using a High-Precision, Low-Overhead Embedded Test Structure", *Hardware-Oriented Security and Trust*, 2012, pp. 75-82.
[46] M. Li, A. Davoodi and M. Tehranipoor, "A Sensor-Assisted Self-Authentication Framework for Hardware Trojan Detection", *Design, Automation & Test in Europe Conference*, 2012.
[47] D. Du, S. Narasimhan, R. S. Chakroborty and S. Bhunia, "Self-Referencing: a Scalable Side-Channel Approach for Hardware Trojan Detection", Cryptographic Hardware and Embedded Systems, 2010, pp. 173-187.
[48] M. Potkonjak, A. Nahapetian, M. Nelson, and T. Massey, "Hardware Trojan Horse Detection using Gate-Level Characterization", *Design Automation Conference*, 2009, pp. 688-693.
[49] S. Wei, K. Li, F. Koushanfar and M. Potkonjak, "Provably Complete Hardware Trojan Detection using Test Point Insertion", *International Conference on Computer-Aided Design*, 2012, pp. 569-576.
[50] S. Wei and M. Potkonjak, "Malicious Circuitry Detection Using Fast Timing Characterization via Test Points", *Symposium on Hardware-Oriented Security and Trust*, 2013.
[51] B. Cha and S.K. Gupta, "Efficient Trojan Detection via Calibration of Process Variations", *Asian Test Symposium*, 2012.
[52] B. Cha and S. K. Gupta, "Trojan Detection via Delay Measurements: A New Approach to Select Paths and Vectors to Maximize Effectiveness and Minimize Cost", *Design, Automation & Test in Europe*, 2013.
[53] K. Xiao, X. Zhang and M. Tehranipoor, "A Clock Sweeping Technique for Detecting Hardware Trojans Impacting Circuits Delay", *Design & Test*, Vol. 30 Issue 2, 2013, pp. 26-34.
[54] Y. Liu, K. Huang, Y. Makris, "Hardware Trojan Detection through Golden Chip-Free Statistical Side-Channel Fingerprinting", *Design Automation Conference*, 2014, pp. 1-6.
[55] N. Yoshimizu, "Hardware Trojan Detection by Symmetry Breaking in Path Delays", *International Symposium on Hardware-Oriented Security and Trust*, 2014, pp. 107-111.
[56] S. Deyati, B. J. Muldrey, A. Singh and A. Chatterjee, "High Resolution Pulse Propagation Driven Trojan Detection in Digital Logic: Optimization Algorithms and Infrastructure", *Asian Test Symposium*, 2014, pp. 200-205.
[57] I. Exurville, L. Zussa, J.-B. Rigaud and B. Robisson, "Resilient Hardware Trojans Detection based on Path Delay Measurements", *International Symposium on Hardware-Oriented Security and Trust*, 2015, pp. 151-156.
[58] I. Wilcox, F. Saqib, and J. Plusquellic, GDS-II Trojan detection using Multiple Supply Pad V_{DD} and GND I_{DDQ}s in ASIC Functional Units", *International Symposium on Hardware-Oriented Security and Trust*, 2015.

[59] D. Ismari, C. Lamech, S. Bhunia, F. Saqib and J. Plusquellic, “On Detecting Delay Anomalies Introduced by Hardware Trojans”, *International Conference on Computer-Aided Design*, 2016.
[60] W. Che, M. Martin, G. Pocklassery, V. K. Kajuluri, F. Saqib and J. Plusquellic, “A Privacy-Preserving, Mutual PUF-Based Authentication Protocol”, *Cryptography*, Vol. 1, Issue 1, 2016.
[61] S. Narasimhan, D. Du, R. S. Chakraborty, S. Paul, F. Wolff, C. Papachristou, K. Roy and S. Bhunia, “Multiple-Parameter Side-Channel Analysis: a Non-Invasive Hardware Trojan Detection Approach”, *International Symposium on Hardware-Oriented Security and Trust*, 2010, pp. 13-18.