6-9-2016

DETECTING DELAY ANOMALIES INTRODUCED BY HARDWARE TROJANS USING CHIP AVERAGING AND AN ON-CHIP HIGH RESOLUTION EMBEDDED TEST STRUCTURE

Dylan Ismari

Dylan Ismari
Candidate

Computer Engineering
Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

Jim Plusquellic, Chairperson

Payman Zarkesh-Ha

Fareena Saqib

Raj Chakraborty

DETECTING DELAY ANOMALIES INTRODUCED BY HARDWARE TROJANS USING CHIP AVERAGING AND AN ON-CHIP HIGH RESOLUTION EMBEDDED TEST STRUCTURE

BY

DYLAN ISMARI

B.S., Computer Engineering, University of New Mexico (UNM), 2012

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

Engineering

The University of New Mexico
Albuquerque, New Mexico

May, 2016

DEDICATION

This dissertation is dedicated to everyone who has helped me along the way and the life that I have yet to live.

ACKNOWLEDGMENTS

I would like to to express my greatest gratitude to my advisor and committee chair Dr. Jim Plusquellic. Without his encouragement and guidance, I would not have been able to achieve my goals. He has been a great inspiration.

I would also like to thank my committee members Dr. Payman Zarkesh-Ha, Dr. Fareena Saqib, and Dr. Raj Chakraborty for their involvement and support. I am also grateful for the community members in the Electrical and Computer Engineering Department at the University of New Mexico continuous direction and aid. I would lastly like to thank the group members of our research group, specifically Wenjie Che, Mitch Martin, Ian Wilcox, and Bill Cavanaugh, for their weekly input and timeless contributions.

DETECTING DELAY ANOMALIES INTRODUCED BY HARDWARE TROJANS USING CHIP AVERAGING AND AN ON-CHIP HIGH RESOLUTION EMBEDDED TEST STRUCTURE

by

Dylan Ismari

B.S., COMPUTER ENGINEERING,
UNIVERSITY OF NEW MEXICO (UNM), 2012
PH.D., ENGINEERING,
UNIVERSITY OF NEW MEXICO (UNM), 2016

ABSTRACT

A hardware Trojan (HT) detection method is presented that is based on measuring and detecting small systematic changes in path delays introduced by capacitive loading effects or series inserted gates of HTs. The path delays are measured using a high resolution on-chip embedded test structure called a time-to-digital converter (TDC) that provides approx. 25 ps of timing resolution. A calibration method for the TDC as well as a chip-averaging technique are demonstrated to nearly eliminate chip-to-chip and within-die process variation effects on the measured path delays across chips, which simplifies the process and enhances the effectiveness of a simulation-based golden model. Path delay tests are applied to multiple copies of a 90 nm custom ASIC chip which incorporates two copies of an AES macro as well as a SPICE-level transient simulation model. The AES macros are exact replicas except for the insertion of several additional gates in the second hardware copy, which are designed to model HTs. Statistical detection methods are used to isolate and detect systematic changes introduced by these additional gates and a set of Trojan emulation circuits also inserted into the macros.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

Chapter 1
Introduction

1.1 Hardware Security

Hardware security has become a prominent forum of research over the past decade due to increasing segmentation of the design, manufacturing, and testing of integrated circuits (ICs). There has emerged two focuses in the field which are Hardware Trust and Physical Unclonable Functions (PUFs) design. A focus of the Hardware Trust has been Hardware Trojan detection. Hardware Trojan detection gets its namesake from the story of the Trojan Horse of Ancient Greece. The analogue of the story to the area of hardware security is having unknown and unwanted extraneous circuitry added to a design with malicious intent. Physical Unclonable Functions have arose as devices to aid encryption, authentication, true-random-number generation, and device identification. The main research concern of PUFs has been creating cryptographically strong security primitives for use in encryption. The work for this proposal has been in both areas. The continued research of this work will be in Hardware Trojan detection.

1.2 Hardware Trojan Detection Introduction

Hardware trust has emerged as a major concern for government and industry personnel, as made evident from a wide variety of issues raised at recent technical meetings [1][2]. Unlike the PUF research of hardware security which provides a 'value add' to products, hardware trust is something that customers expect, similar to the expectations they have regarding manufacturing defects. Unfortunately, providing a high assurance, trusted product is much more difficult than providing high quality, defect-free chips. This is true because the random nature of manufacturing defects makes it possible to find nearly all of them with test vectors that provide high levels of fault coverage. Hardware Trojans (HTs), on the other hand, are designed and inserted by intelligent adversaries with the deliberate intention of making them nearly impossible to activate with arbitrary test vectors, akin to the difficulty of guessing a 256-bit AES key using only information provided by a chosen message analysis. However, there are alternative techniques for detecting HTs that do not require activation. Drawing on the analogy with AES, a technique called differential power analysis greatly simplified the task of stealing the AES key. Similarly, parametric testing methods provide a similar advantage for detecting HTs by carrying out a structural analysis of the IC, as opposed to a functional analysis. The advantage of a structural analysis over a functional analysis relates to the number of tests that need to be applied to attain sufficient coverage of HTs. Structural testing, as the name implies, focuses on testing each of the elements in the netlist or layout, and therefore, the number of tests are related linearly to the size of the circuit. In contrast, functional analysis requires an exponential number of test vectors, which is not practical except for very small chips. Any parameter of the IC, including dynamic current, leakage, delay, EMI, hot spots, etc. can be targeted by parametric methods. Delay-based parametric methods detect delay anomalies introduced by the capacitive loading of HT wires and by series inserted HT gates, which is leverage in our research. In contrast to random defects, the anomalies introduced by HTs are systematic in nature, i.e., showing up in multiple copies of the ICs in a similar fashion, and can be identified by comparing the signal behavior of the chips with that of a golden (HT-free) simulation model. The challenge of implementing parametric testing methods is dealing with chip-to-chip and within-die process variations effects. Failing to properly account for the natural variations that occur in the power and performance characteristics of chips results in false negative decisions (a determination that the chip does not have an HT when it does) and false positive HT decisions (a decision that it has an HT when it does not).

1.3 PUF Introduction

Physical unclonable functions (PUFs) are hardware security primitives designed to produce random but reproducible bitstrings from variations in the printed and implanted features of wires and transistors on an integrated circuit (IC). Each IC is uniquely characterized by random manufacturing variations, and therefore, the bitstrings are unique from one chip to the next. PUFs can serve several important security applications including authentication and cryptography, which in turn can be used for secure communications, anti-counterfeiting, detecting malicious system alterations in the field, feature activation, hardware metering, etc.

1.4 Dissertation Organization

What follows is an account of research done for use within this dissertation. Chapter 2 dives into the work completed in PUFs that could not be continued due to Non-Disclosure Agreements that were set post-manufacturing. Chapter 2 covers a proposed system with simulation results and statistical analysis of the design. Chapter 3 starts the Hardware Trojan detection research and the bulk of this dissertation introducing a method of Trojan detection. The 90nm chip used for data collection is described with a description of the time-to-digital converter (TDC). Chapter 4 looks into the TDC sensitivity analysis by leveraging HT Emulation Circuits (HTECs). The techniques used to account for chip-to-chip variations and within-die variations, as well as a selection criteria for which measurements were to be used finish Chapter 4. Chapter 5 looks into the experimental results of the HTECs and a set of fan-out and series inserted gates. Regression detection analysis is used in conjunction with a trending detection analysis and chip-averaging. Chapter 6 details the simulation based work done to verify and complement the hardware experiments. Path coverage analytics are also discussed. Chapter 7 discusses possible future work with chapter 8 concluding this dissertation.

Chapter 2
JellyFish PUF

2.1 Proposed PUF System

In this proposal, we present a full on-chip implementation of a PUF system that we call the JellyFishPUF (**JFP**). The entropy source leveraged in the PUF is based on within-die variations in transistors and conductors, e.g., polysilicon, metal wires, vias and contacts, and is similar in structure to a living jellyfish. An array of 2,048 identically designed cells, called stimulus/measure circuits or SMCs, defines the entropy source. The voltages produced by the SMCs are routed using pass gates to an on-chip voltage-to-digital-converter or VDC. The VDC converts the SMC voltages into 8- bit digital values, which reflect their relative magnitudes. A digital controller accepts inputs from user applications and carries out the specified functions, e.g., enrollment for secret key generation. The output of the JFP engine is a 256-bit (or larger) bitstring plus public data to handle functions that require regeneration. The contributions of this work are as follows:

- A complete PUF engine, implemented in a 65nm technology, with area 0.125mm^2 and 256-bit bitstring generation time as small as 2ms.
- The implementation of bitstring functions including enrollment, regeneration, authentication and random number generation, as well as a new function call “band enrollment” which provides of the order of 2^n unique 256-bit bitstrings per chip.
- An implementation of a process called **normalization** that eliminates

transistor-based variations from the entropy source but preserves resistance variations in the conductors, i.e., polysilicon, metal wires, contacts, and vias. Our analysis shows this process improves reliability of the bitstring regeneration process due to the harder to anticipate variations in transistors being removed.

- The implementation of a differential power analysis resistant VDC, important to fend off key extraction based upon the digitization process, and a process called **calibration** that allows the VDC to digitize voltages from the SMC across temperature variations from -40°C to 125°C and supply voltage variations of +/- 10% of nominal.
- An entropy source that leverages single vias, minimum width metal, and polysilicon wires over 5 of the metal layers available in the IP block.

References [3-4] describe preliminary results from a test chip which uses a metal-based entropy source, a “pulse- shrinking” version of the VDC as well as processes related to calibration, thresholding, and XMR. However, this proposal integrates all of these components in a unified system architecture and investigates each of the novel concepts described above.

2.2 PUF Background

Random bitstrings form the basis for encryption, identification, authentication, and feature activation in hardware security. The introduction of the silicon PUF as a mechanism to generate random bitstrings began in [5], although their use as chip identifiers began a couple years earlier[6]. Since their introduction, there have been many proposed architectures that are promising for PUF implementations, including those that leverage variations in transistor threshold voltages [6-7], in delay chains and ring oscillators [5][8-10+many others], in SRAMs [12-13], in leakage current [14], in metal and transistor resistance [3-4], in clock networks [15], in scan chains, [16] and transmission lines. [17]

2.3 Jellyfish PUF

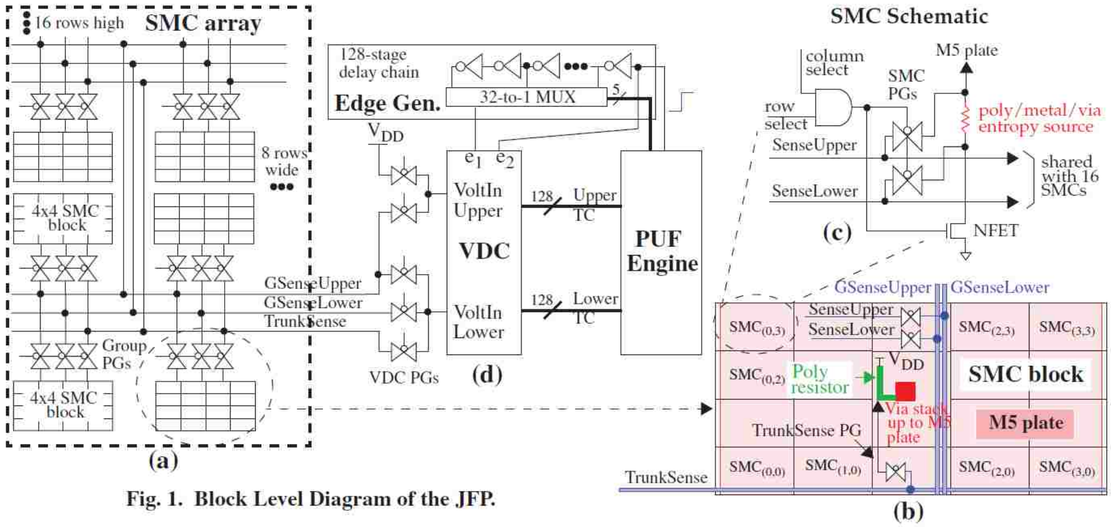

Fig. 1. Block Level Diagram of the JFP.

Figure 1: Block Level Diagram of the JFP.

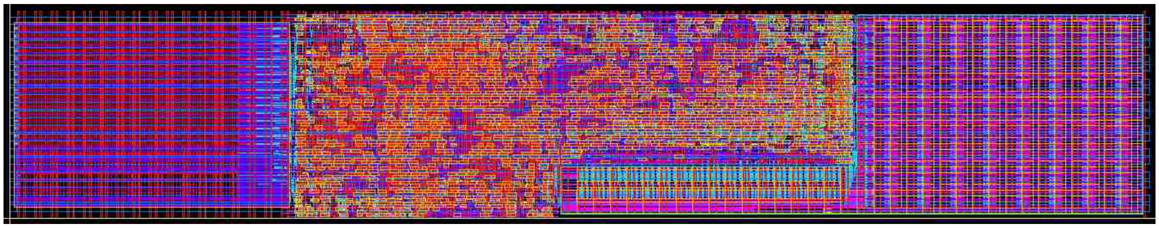

Figure 2: JFP layout in a 65nm technology with dimensions of 830μm x 150μm.

A block level diagram of the JellyFishPUF (JFP) PUF system architecture is shown in Fig. 1 and its layout in Fig. 2. The PUF Engine is a digital controller that coordinates a series of operations described in the following sections. The SRAM shown on the left in Fig. 2 is used by the PUF Engine for Voltage Distribution analysis and for storing helper data during bit generation. The edge generator (Edge Gen.), voltage-to-digital converter (VDC) and SMC array define the entropy source and conversion components.

The SMC (stimulus-measure-circuit) array (Fig. 1(a)) is defined as a set of 128 4x4 SMC blocks, arranged in 16 rows and 8 columns. Each SMC element within a 4x4 block (Fig. 1(b)) is able to provide a single component of entropy. Therefore, the entire array defines an entropy source with 2,048 components.

Fig. 1(c) shows a schematic of an SMC element. It is composed of an AND gate, which serves to enable the SMC, and two pass gates (PGs) connected across the entropy source. The entropy source is a 1µm silicided-polysilicon wire and a single via stack from poly up to M5. An NFET connects to the polysilicon wire and provides the stimulus of approx. 500µA when the SMC is enabled. The M4-M5 via on the upper end of the entropy source connects to an M5 metal plate that covers the 4x4 SMC block as shown in Fig. 1(b). The plate is connected to the VDD supply grid through a controlled-resistance silicided-Poly resistor of approx. 400 Ohms. Therefore, when an SMC is enabled, the NFET current creates a voltage drop across the entropy source which can be sensed by the two PGs. The M5 plate and Poly resistor provide a common node connected to VDD for all SMCs in the 4x4 block. This common node in combination with the TrunkSense PG shown along the bottom of Fig. 1(b) allow voltage variations introduced by the different NFET currents within the SMCs of the block to be eliminated. The process, called normalization, is described below.

The SMC PGs connect to two wires labeled SenseUpper and SenseLower, which are shared among all SMCs within the block. Two additional PGs at the block level connect these wires to two globally routed GSenseUpper and GSenseLower wires shown in Fig. 1(a) and (b), which connect across the 128 SMC blocks (the same is true of the TrunkSense wires). These wires route out of the SMC array to three VDC pass gates shown in Fig. 1(d). The PUF Engine provides a sequence of control signals which allow search of these sense voltages to be digitized by the VDC.

The inputs of the VDC are two voltages labeled VoltInUpper and VoltInLower and two wires e1 and e2 that are connected to the Edge Gen. (Fig. 1(d)). The VDC outputs two 128-bit thermometer codes (TCs) that reflect the magnitude of the sense voltage inputs. A TC is defined as a string of 0's (or 1's) followed by a string of 1's (or 0's).

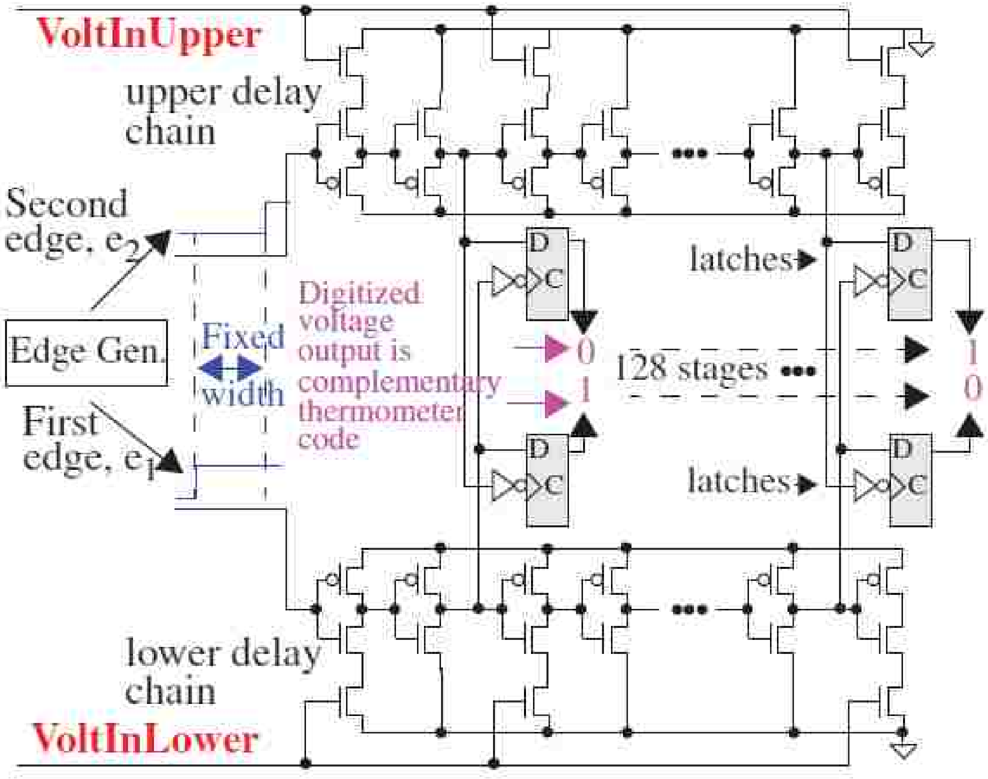

Figure 3: Voltage-to-Digital Converter (VDC).

Fig. 3 gives a schematic of the VDC to better illustrate the digitization process.

The VDC is composed of two 256-stage delay chains. The VoltInLower input connects to 128 NFETs, inserted in series with the odd-numbered inverters in the delay chain. VoltInUpper connects in a similar fashion to the upper delay chain. The PUF Engine starts the digitization process by driving a rising edge into the EdgeGen. as shown in Fig. 1(d). The Edge Gen. passes e1 to the corresponding VDC input but delays e2 by a Δt (determined by 32-to-1 select MUX). The two edges then ‘race’ down the two inverter chains at speeds relative to the magnitude of the VoltInUpper/Lower inputs.

Under the condition that VoltInUpper > VoltInLower, the edge propagating along the top delay chain eventually passes the edge on the bottom delay chain. The outputs of the even inverters along both delay chains connect to a set of latches that record the point at which this occurs. As shown in Fig. 3, the TC produced by the latches on the upper chain is a sequence of 0’s followed by 1’s, while a complementary pattern appears on the latch outputs of the lower chain. A value proportional to the magnitude of the voltage difference between VoltInUpper and VoltInLower can be obtained by counting the number of 1’s in either of these TCs. We refer to the number of 1’s as a **VDCNum**.

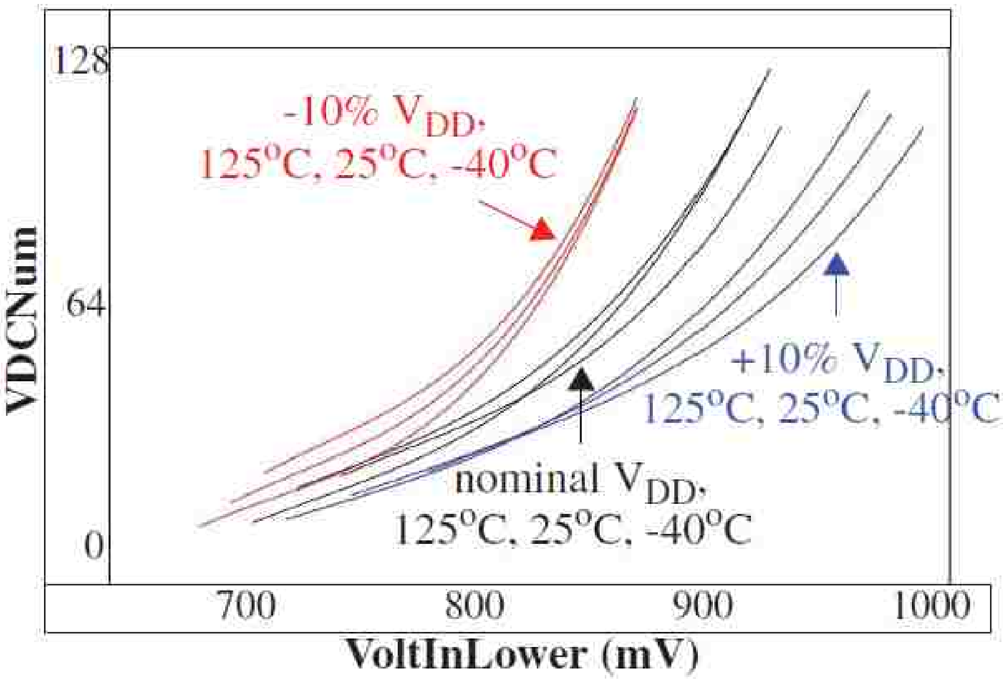

Figure 4: VDC calibrated transfer curves at 9 temperature/voltage corners under the TT Nominal process curves.

The transfer curves in Fig. 4 illustrate the behavior of the VDC across 9 temperature/voltage (TV) corners. The data for the curves is generated by SPICE-level simulations of an RC layout-extracted model of the Edge Gen. and VDC components of Fig. 2 under typical transistor (TT) and nominal wire resistance and capacitance conditions (nominal). The data for the curves is generated by fixing VoltInUpper at V_{DD} and sweeping the voltage on VoltInLower from 650mV to 1.0V, in 20mV steps. The Δt between the edges e1 and e2 is fixed for any one curve but is "tuned" for each TV corner. A **calibration process** implemented within the PUF Engine determines the appropriate Δt by monitoring the VDCNums produced with the VoltInLower set to the largest value on the curve. The control inputs to the 32-to-1 MUX are set such that the VDCNum produced under this condition is less than the overflow value of 128. Calibration involves successively reducing the Δt by reducing the digital select inputs of the 32-to-1 MUX.

The transfer curves illustrate that the mapping from voltages to VDCNum is non-linear. In particular, the VDC has higher sensitivity to changes in VoltInLower at the high end of the voltage range, than at the lower end. Although this “distorts” the Gaussian nature of the voltage variations produced by the entropy source, our proposed voltage comparison technique (described below) is able to provide unbiased random values from the distribution.

2.4 PUF Engine Processing Techniques

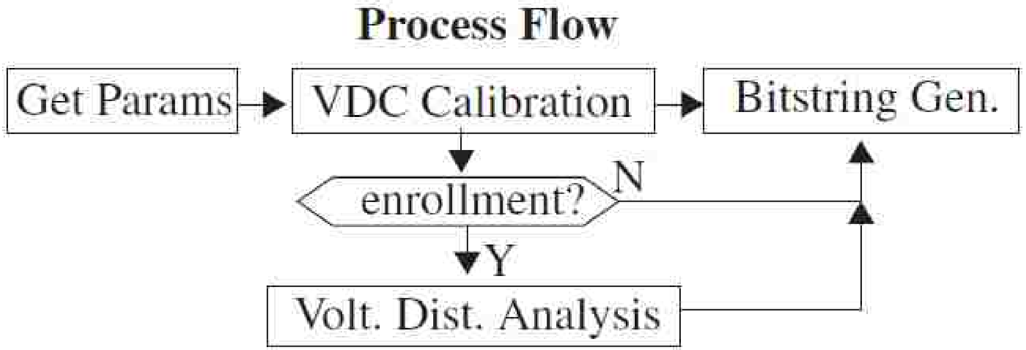

Figure 5: Overall flow of bit generation process.

The overall flow of the bit generation process is given in Fig. 5. User parameters, such as function and the size of the bitstring to generate, are stored in memory-mapped registers while the PUF Engine remains in idle mode. A “bit gen start” input to the PUF Engine starts the process by first performing calibration. The voltages used during calibration are a user-selectable subset of those produced by the SMC array itself, which ensures that overflow does not occur during subsequent processes.

Voltage distribution analysis is the second process to be performed when the function is enrollment or band enrollment (described below). This process constructs a histogram of digitized voltages from a subset of SMC elements and derives a value that reflects its width, which is distinct for each chip because of global variations in process parameters. Our proposed **thresholding technique** decides which voltage comparisons are “stable” enough to generate a bit and makes use of this width information to improve reliability. The bitstring generation process follows the calibration and voltage distribution analysis processes.

2.4.1 Sample Analysis

User input parameters can be used to specify the number of samples that are averaged for each of the three processes, as a means of reducing measurement noise. As indicated, the VDC produces a value between 0 and 128 (8-bits). In order to fully leverage the benefits of averaging, the VDCNums are scaled to **11-bits**, with the 3 low-order bits representing 3 binary digits of precision. For example, the average value produced when 3 samples of 75 and 5 samples of 76 are generated by the VDC is 605 which is 75.625 in fixed point.

2.4.2 Digitization Options

The current sourced by the NFET within the SMC creates a voltage drop across the entropy source. The GSenseUpper and GSenseLower sense wires transfer this voltage to the VDC inputs through PGs, as discussed above. We use voltage drops as the entropy source because they eliminate bias effects that can occur for SMCs in different regions of the array. Voltage drops are defined as ($V_{GSenseUpper}$ - $V_{GSenseLower}$). The control inputs to the VDC PGs shown in Fig. 1(d) provide two options for digitizing the voltage drops, which are referred to as the **digital** and **analog** methods. For the digital method, each of

$V_{GSenseUpper}$ and $V_{GSenseLower}$ are digitized separately using the VoltInLower input of the VDC and the voltage drop is computed digitally from the 11-bit VDCNums. The VoltInUpper input in this case is set to V_{DD}. In contrast, the analog method places GSenseUpper on the VoltInUpper input and GSenseLower on the VoltInLower input. In this case, the single VDCNum produced reflects the difference in the analog voltages directly.

2.4.3 Normalization

The primary reason for partitioning of the SMC array into a set of 128 blocks is to support a special process called normalization. The objective of **normalization** is to eliminate transistor current variations as a component of the measured voltage drops across the entropy stack. Previous work suggests that the current-induced variations contribute significantly to TV noise, which, in turn, acts to reduce the probability of correctly regenerating the bitstring [13].

$$R = \frac{\left(V_{GSenseUpper} - V_{GSenseLower} \right)}{I_{NFET}} \quad \textbf{Eq. 1.}$$

$$R_{norm} = \frac{\left(DV_{GSenseUpper} - DV_{GSenseLower} \right) * 256}{\left(129 - DV_{TrunkSense} \right)} \quad \textbf{Eq. 2.}$$

We think of normalization as a process that ‘normalizes’ the voltage drops for all SMCs within the block to a reference current. Normalization is derived from the basic circuit theory equation R = V/I given by Eq. 1 which states that the resistance of the entropy source can be obtained from the sense voltage measurements by dividing through by the NFET current. Unfortunately, measuring currents on-chip is challenging and impractical. Eq. 2 provides an alternative in cases where it is only necessary to determine a value that is “proportional” to resistance. Here, $DV_{TrunkSense}$ is the digitized voltage (a value between 0 and 128) produced at the point where the Poly resistor connects to the M5 plate, as shown in Fig. 1(b). Current from any of the enabled SMCs in the block must flow across the Poly resistor and past this point on the M5 plate. Therefore, the voltage drop defined by (129 - $DV_{TrunkSense}$) is proportional to the NFET current because the Poly resistor is a shared connection to V_{DD} for all elements within the SMC block.

Normalization is a user specified option, that when enabled, instructs the PUF Engine to additionally digitize $V_{TrunkSense}$ and to carry out the operations defined by Eq. 2. The multiplication factor of 256 scales the digitized voltage drop and allows the result, R_{norm}, to be expressed and manipulated as an integer in the bit generation process.

2.4.4 Thresholding

Thresholding is used to improve the reliability of the bit generation process for applications that require exact regeneration of the same bitstring under different TV conditions. Bit generation options called enrollment and a new one called band enrollment use thresholding to accomplish this goal. Thresholding requires the difference between a pair of VDCNums obtained from two distinct SMCs to exceed a threshold. The threshold is derived by multiplying a user-specified value between 0.0 and 1.0 with the range of the Voltage Distribution computed earlier.

2.4.5 Bit Generation Options

The bit generation process compares the digitized voltage **drops** from a sequence of (SMC_x, SMC_y) pairings. The sequence is determined by two user-specified seeds and corresponding linear feedback shift registers (LFSRs). The LFSR sequencing with normalization enabled restricts comparisons to within each SMC block. A '1' bit is generated if the VDCNum of SMC_x in the pairing is greater than the VDCNum of SMC_y, otherwise a ‘0’ is generated.

The PUF Engine allows the user to specify one of five different bit generation options, including enrollment, band enrollment, regeneration, authentication, and true-random-number generation (TRNG). The difference in the process of generating a bitstring under each of these options is related to how two threshold parameters are used.

The distributions shown in Fig. 6 are used to illustrate the bit generation options. These distributions, unlike the distribution used in Voltage Distribution analysis, are defined using VDCNum **differences**, i.e., ($VDCNum_{SMCa}$ - $VDCNum_{SMCb}$) obtained from a pairing of SMCs a and b. The differences are plotted along the x-axis against their frequency of occurrence on the y-axis in the plots.

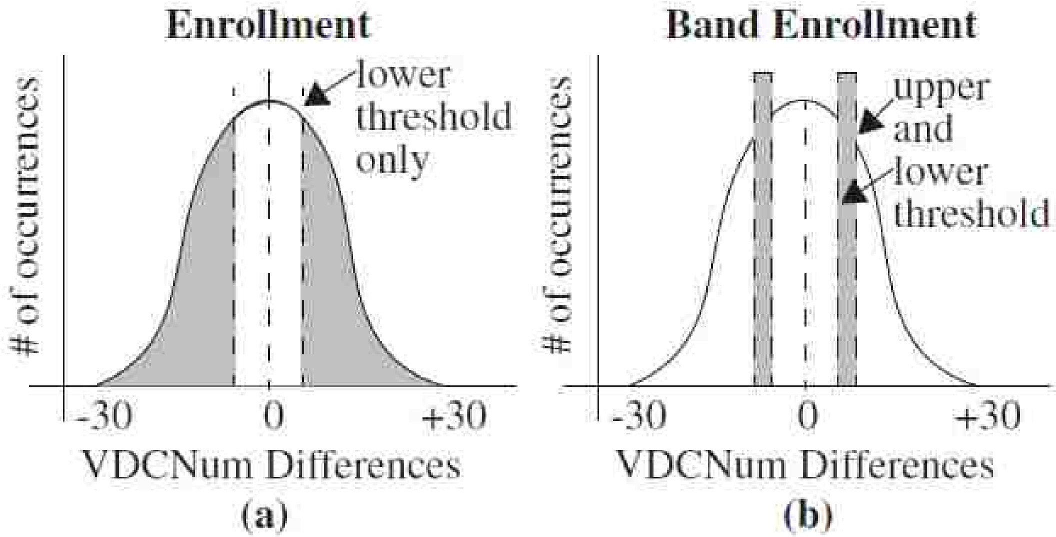

Figure 6: Thresholding under (a) Enrollment and (b) Band Enrollment.

The thresholding criteria used for **enrollment** is shown in Fig. 6(a). The shaded regions on both sides of the distribution, delineated by “lower threshold only”, represents comparisons whose differences exceed the threshold, and therefore are permitted to generate a bit.

Band enrollment is similar and is illustrated in Fig. 6(b). A second objective of band enrollment is to provide a unique bitstring each time enrollment is carried out when the same seed is used for the LFSRs. This occurs to some degree for the enrollment function because VDCNum differences that are close to the threshold can be on either side of it during any given enrollment process. This is true because the statistical nature of measurement noise introduces uncertainty in the VDCNum differences. Band enrollment simply increases the probability closer to 50% that any VDCNum difference can be inside or outside the band during an enrollment process. It accomplishes this by using a second user-specified “upper” threshold to create two narrow bands as shown in Fig. 7.

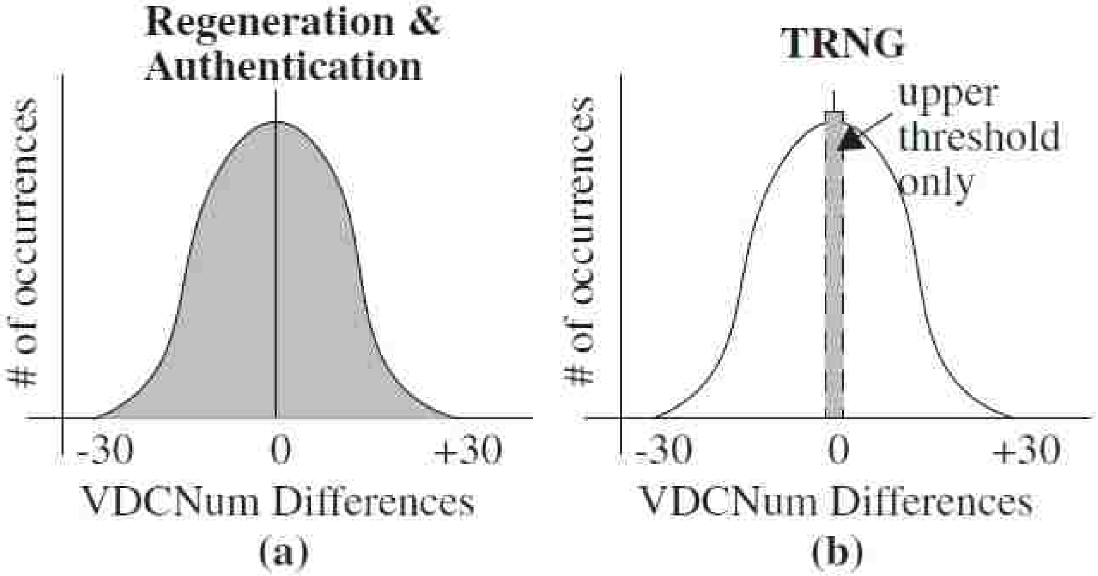

Figure 7: Thresholding under (a) Regeneration/Authentication and (b) True-Random-Number-Generation Modes.

The **regeneration** and **authentication** functions do not use thresholding and therefore all comparisons are valid, as illustrated by the shaded region in Fig. 7(a). However, the actual comparisons used during regeneration would be chosen in most cases from the shaded regions as shown in Fig. 6(a). This is true because the valid comparisons used in regeneration are determined from a “helper data” bitstring, which is stored in the SRAM shown in Fig. 2 during enrollment. Enrollment records a ‘1’ when a comparison is valid and a ‘0’ when it is not. Regeneration reads this bitstring to ensure the same sequence of comparisons are carried out.

The valid band for true-random-number-generation (**TRNG**) is narrow and centered around the mean of the distribution as shown in Fig. 7(b). In this region, noise sources readily change the sign of the VDC difference and therefore the bit value varies randomly even for the same sequence of comparisons. The lower threshold is set to 0 and only the upper threshold is used to accomplished this. Note that band enrollment (without regeneration) can also be used as a TRNG.

2.4.6 Reliability Enhancing Techniques

In addition to the normalization technique described above, the PUF Engine also implements a method called **XMR** for increasing the probability of correctly regenerating the bitstring [4]. XMR creates an odd number of “copies” of the bitstring during enrollment and regeneration. A bit-wise majority voting technique is then carried out column-wise for each bit across the n copies during regeneration as a means of preventing bit flips in the final bitstring. For example, if the user specifies that 3 copies of the bitstring are to be generated (called 3MR), the majority voting scheme can “correct” single bit flips that occur in any column. For 5MR, two bit flips per column can occur while preserving the ability to correctly regenerate the bitstring, and so on. We investigate these reliability enhancing schemes in the simulation experiments described below.

2.5 Experimental Results

Over 100K SPICE-level Monte Carlo simulations were run on an RC layout-extracted model of the SMC block shown in Fig. 1(b) under each of the 9 TV corners shown for the VDC transfer curves in Fig. 4. This allowed us to model 50 instances of the SMC array, i.e., 50 instances*2048 SMCs. In addition to the transistor level variations introduced by the foundry models, we modified the RC netlist to enable CADENCE Spectre to introduce within-die variations in the poly, metal wire, contacts, and vias of the SMC block. The Monte Carlo sample limits for within-die variations of these components were set to approx. 30% of the chip-to-chip variations specified in the design manual.

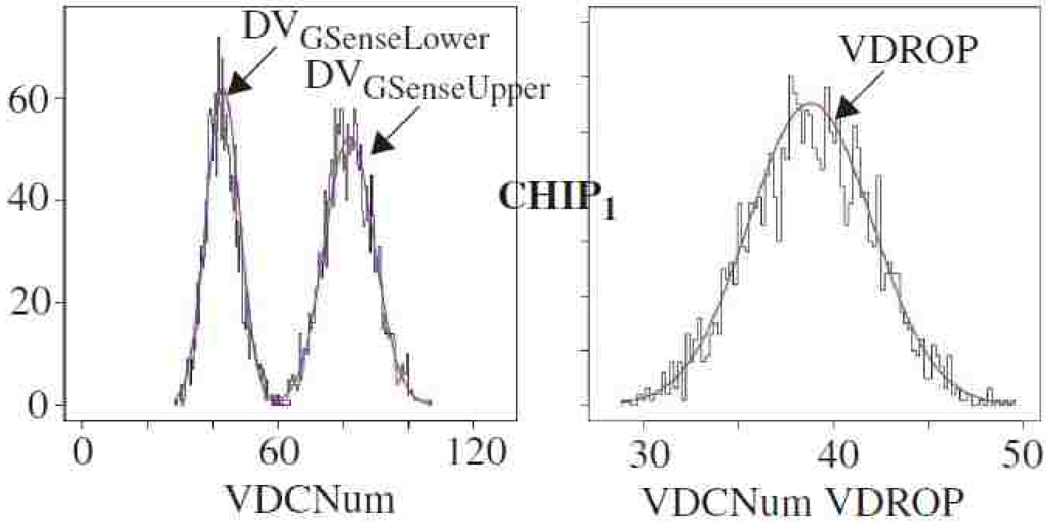

Figure 8: (a) Digitized GSenseUpper/Lower distributions from TT Nominal, nominal V_{DD}, 25ºC MC simulations, (b) and corresponding VDCNum differences.

The VDCNums corresponding to the $V_{SenseUpper}$ and $V_{SenseLower}$ values obtained from the MC simulations are derived using the transfer curves given in Fig. 4. Fig. 8(a) gives the distributions of the digitized voltages for $CHIP_1$, labeled $DV_{GSenseLower}$ and $DV_{GSenseUpper}$. Although not shown, the distribution using the actual voltage, $V_{GSenseLower}$, is wider than the distribution for $V_{GSenseUpper}$, and is opposite to the behavior shown in Fig. 8(a) where $DV_{GSenseUpper}$ is wider. The changing slope of the transfer curve changes the spread of the two distributions.

Fig. 8(b) shows the distribution of the VDCNum VDROPs, which is derived from a pair-wise subtraction of the DV_{GSense} values from Fig. 8(a). Although the distribution appears to be normal, an analysis of all 50 chips shows that the percentage of values above and below the mean varies by up to 2%, i.e., the worst chip distribution has 48% of the values below the mean and 52% above. This is also an artifact that is introduced by the non-linear transfer curve. However, our comparison methodology selects pairs of values randomly from this distribution to create VDCNum differences during bit generation, and as a consequence, **it is robust to non-Gaussian shapes in the underlying distributions**.

2.5.1 Randomness, Uniqueness, and Reliability Analysis

Inter-chip hamming distance (HD) and the NIST statistical tests [19] are used to evaluate the statistical quality of bitstrings of size 512 bits generated under three XMR scenarios, including 0MR, 3MR and 5MR, each with (N) and without (NN) normalization enabled. The user-specified threshold in this analysis is set such that no bit flips occurred (intra-chip hamming distance is 0) across any of the 50 chips. The Inter-chip HDs given in Table 1 indicate that the uniqueness of the bitstrings are close to the ideal of 50%.

	0MR N	0MR NN	3MR N	3MR NN	5MR N	5MR NN
Inter HD	50.02%	50.13%	50.14%	50.04%	49.98%	50.05%

Table 1: Inter-Chip HDs under 6 usage scenarios.

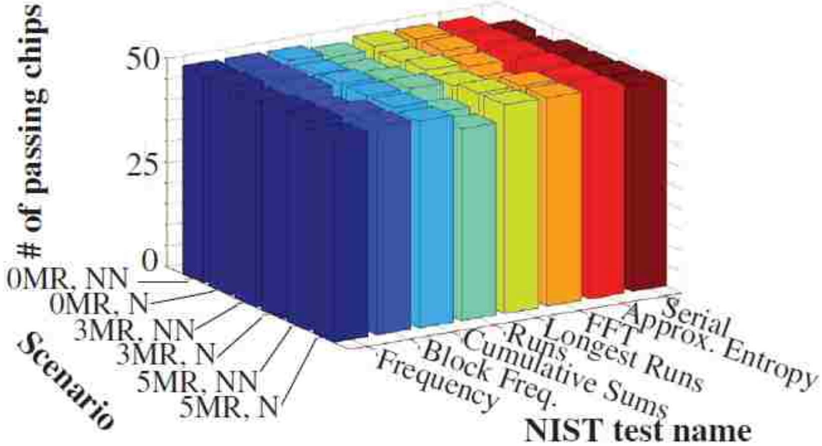

Figure 9: NIST Test Results

Fig. 9 depicts the results of the NIST tests at a significance level of 0.01. The pass criteria for each test is that 47 or more of the chips pass the test individually. All tests are passed except two of the Serial tests, which had 45 and 46 chips passing, and a Runs test with 46 passing. These results indicate that the bitstrings possess a high degree of randomness.

Without normalization, the average intra-chip HD with thresholding disabled is approx. 8% but drops to 4% with normalization, which demonstrates the reliability enhancement capability of normalization.

The reliability enhancing techniques are evaluated by computing a **probability of failure** (POF) metric. A failure is counted when a bit flip occurs in one (or more) of the 8 regenerated bitstrings. A POF is computed by dividing the total number of fails that occur across all chips by the total number of bits in the enrollment bitstrings, which is 50 chips * 512-bits = 25,600. Enrollment is carried out at nominal VDD, 25°C and regenerations at the 8 remaining TV corners.

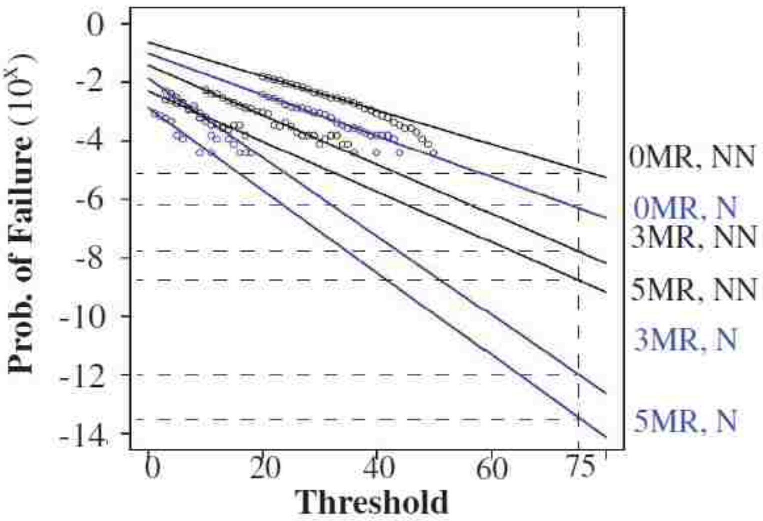

Figure 10: Probability of Failure results for bitstrings.

The user-specified threshold parameter is a value between 0 (no threshold) and 256 (large threshold). We found that a threshold of 75 prevents bit flips in all chips under each of the six scenarios. In order to predict the POF at this threshold, a sequence of bitstring generation experiments are carried out using successively smaller thresholds. The set of POF's from these experiments defines a curve that is exponential in shape. The POF curves and exponential line fits for each of the six scenarios are plotted on a $\log_{10}$ scale in Fig. 10. Horizontal dotted lines provide estimates of the POFs at threshold 75, which vary from $1x10^{-5}$ to $4x10^{-14}$.

The progression of the line fits to steeper negative slopes clearly indicates that both XMR and normalization improve reliability. The main trade-off for higher reliability in either case is bitstring generation time. For example, enrollment bitstring generation time for 0MR, N is approx. 4ms for a bitstring of size 512 and a sample size of 1, and 1ms for regeneration (using the analog difference described above cuts these times in half). In contrast, this time increases to approx. 180ms for enrollment and 45ms for regeneration under 5MR, N with 8 sample averaging.

2.6 JFP Conclusions

A complete IP-level implementation of a resistance-based PUF engine is used to demonstrate several novel PUF features including the entropy source, band enrollment, normalization and calibration, and to analyze important implementation metrics including bit generation time, power consumption, and area overhead. Data from SPICE-level simulation experiments is used to show a high level of statistical quality in the generated bitstring with respect to intra-chip and inter-chip hamming distances, NIST statistical tests, and probability of failure metrics. Due to non-disclosure agreement stipulations, this project has stopped being actively pursued. The preliminary research direction shifted from PUFs to hardware Trojan detection.

Chapter 3
Hardware Trojan Detection

In the rest of this proposal, we investigate an HT detection method that analyzes a chip's delay characteristics. An on-chip measurement structure called a time-to-digital (TDC) is used to obtain high resolution (approx. 25ps) measurements of path delay. Chip-to-chip and within-die process variation calibration methods are proposed as a means of improving resilience to false negative and false positive detection decisions. Experiments are carried out on 44 copies of a custom, 90nm test chip which incorporate two instances of Advanced Encryption Standard (AES) macro. The contributions of this work are as follows:

- Application of our proposed detection methods to commercially synthesized implementations of actual functional units and chips.
- Demonstration of a high resolution, embedded time-to-digital converter (TDC) for measuring path delays.
- Calibration of chip-to-chip variations by tuning a control parameter of the TDC, and calibration of within-die variations using a chip-averaging technique.
- Demonstration that systematic delay anomalies can be measured in data measured from two identical copies of a functional unit with fan-out and payload HTs inserted only in the second copy.
- Determination of the resolution limits of our proposed methods.

The reminder of the paper describes the chip design, experimental setup and test chip data analysis, as well as the statistical outlier techniques that are used to detect the systematic delay anomalies introduced by HTs.

3.2 Hardware Trojan Detection Background

The authors of [19] were the first to address the HT issue. They use transient power supply currents to identify HTs in chips. In [20], logical circuit analysis is carried out using automatic test pattern generation (ATPG) after identifying target sites where HTs are most likely to be inserted. In [21], the authors propose adding observability enhancing changes to the least controllable nodes of the circuit. In [22], the authors extend the state space of the circuit by introducing MUXes that can select the Q or $\overline{Q}$ outputs of the flip-flops. The authors of [23] use a dynamically distributed multiprocessing element trust determination scheme that observes the outputs of the elements to determine trust.

For side-channel variation detection, authors propose methods to decrease noise to better measure minimal power draw [24]. Authors look at region-based stimulation and compare global power consumption [25]. Characterization of regions side-channel signature and observing outliers compared to a set of estimation results are proposed in [26]. Another outlier technique is proposed in [27] where comparisons between a golden model and power signals of a chip. The authors of [28] leverage a calibration technique to identify regional current and frequency fluctuations. Reference [29] takes the characterization to gate-level leakage power and delay measurements.

For delay variation detection techniques, [30] demonstrates a shadow register setup to measure finite delays with minimal test vectors. [31] presents an on-chip test structure to measure delays. The authors of [32] discuss a method of on-chip sensors to measure delay and determine HTs without the need of a golden model. [33] uses fault-injection to create clock glitches with variable clock periods to determine a delay fingerprint. [34] presents a scalable method of circuit partitioning to measure the delay at gate-level. The authors of [35] look to the foundry approximated models to determine expected delay variations. [36] creates a path delay fingerprint of HT-free chips which can be used against chips under question of containing HTs. In [37], the authors look at register-to-register delay of a large number of paths to determine if an HT has been inserted. [38] presents a method to use clock sweeping to obtain the path delay signatures. The authors of [39] propose a self-referencing technique that compares behaviors of identical functional units, thereby eliminating the need for a golden simulation model. This technique is expanded on in [40] by identifying path symmetries and comparing the delays across multiple instances of the path on the same chip. We utilize a similar trending method in this paper to circumvent the golden simulation model.

Chapter 4
Chip Design and Description

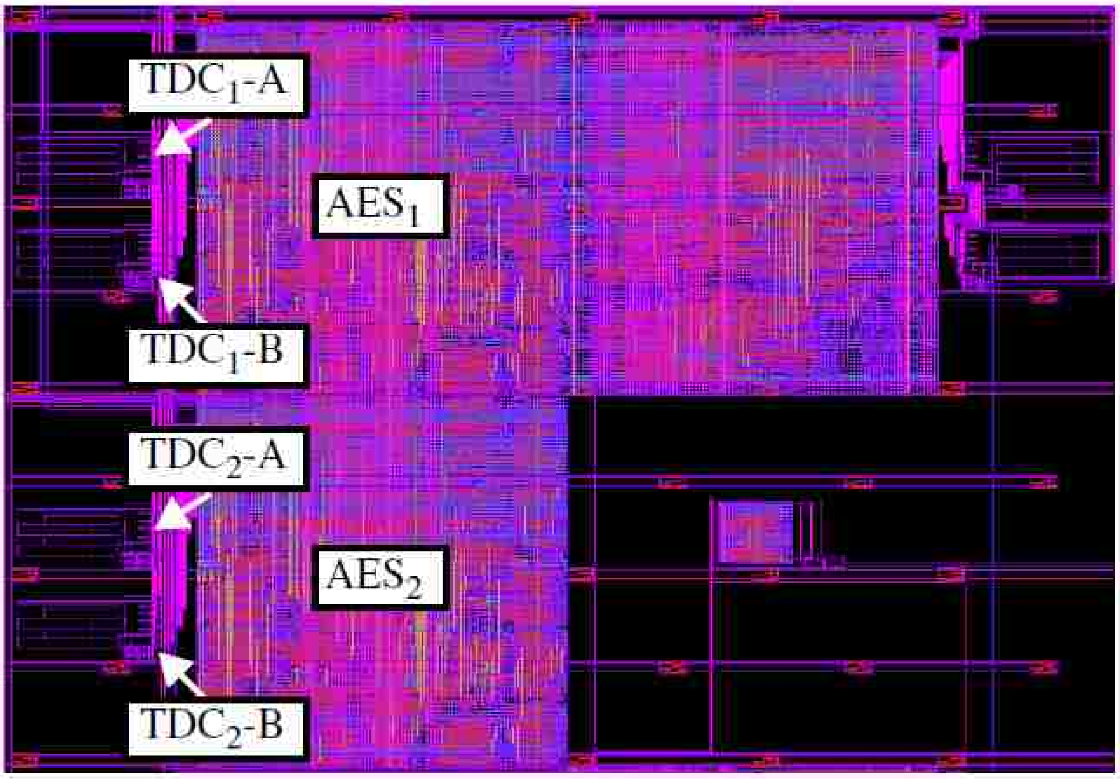

Figure 11: Chip layout showing AES MUTs and TDCs.

The test chip architecture consists of 2 macros-under-test (MUTs), labeled AES$_1$ and AES$_2$ for Advanced Encryption Standard, as shown in the layout of Fig. 11. Two copies of an embedded test structure called the time-to-digital converter or TDC (described below) are associated with each MUT [41]. The TDCs are capable of providing accurate delay measurements of signals propagating through the MUTs.

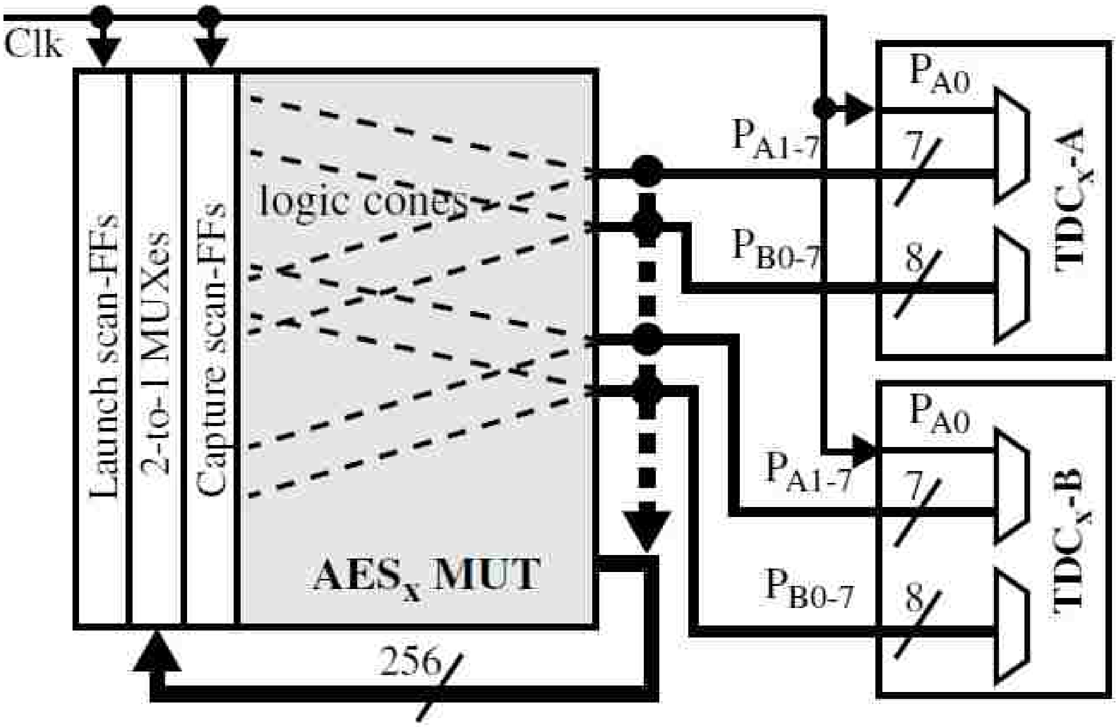

Figure 12: Block diagram of TDC connections to an AES Macro-Under-Test (MUT).

A block diagram illustrating the connectivity between an AES MUT, labeled AES$_x$, and the TDCs labeled TDC$_x$-A and TDC$_x$-B is shown in Fig. 12. Each TDC has 16 inputs. The first input connects to the clock while the remaining 15 connect to outputs of the AES MUT. This allows signals propagating through logic cones to 30 of the AES outputs to be timed by one of the TDCs. The CLK input serves as a means of characterizing the TDCs and as a reference path for delay testing as explained below. A special set of “Launch scan-FFs” are added to the MUT to allow any arbitrary two vector test to be applied. This is accomplished by scanning the first vector into the “Capture scan-FFs” row and the second vector into the “Launch scan-FFs” row. The CLK can then be used to launch transitions onto the inputs of the MUT.

4.1 Time-to-Digital Converter (TDC)

The TDC is designed to measure the relative delay between two output signals from the MUT or one MUT output signal and the CLK.

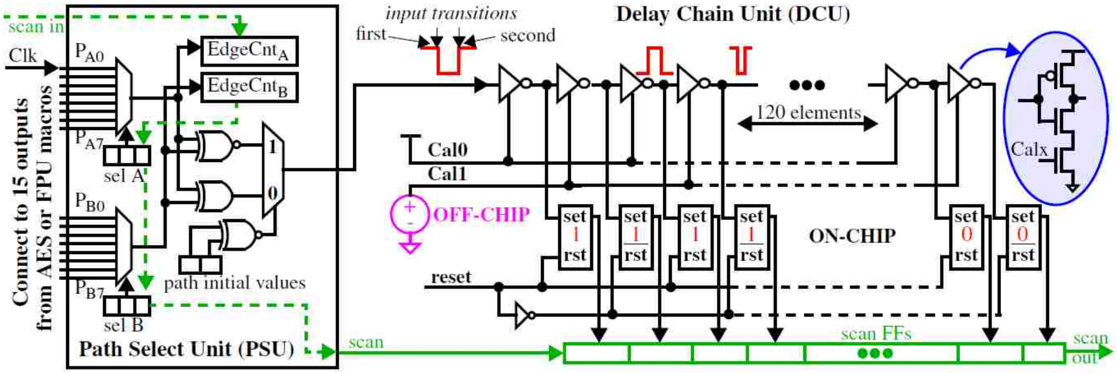

Figure 13: Pulse Shrinking Time-To-Digital Converter (TDC).

The TDC is implemented as two components, labeled Path Select Unit (**PSU**) and Delay Chain Unit (**DCU**) in the schematic of Fig. 13. Scan-FFs in the PSU, labeled “Sel A” and “Sel B”, drive the inputs of two 8-to-1 MUXes, which, in turn, select a specific pairing of MUT outputs, one from the group labeled PA$_x$ and one from group labeled PB$_x$.

Path delay tests are carried out by applying 2 vectors in sequence to the inputs of the MUTs. The output values from the two selected paths to be timed by the TDC after the 1st vector is applied are latched into the “path initial values” control signals. These control values select the output of either the XOR or XNOR gate to generate a negative pulse for the DCU (see Fig. 13). For example, if both path values are 0 or both are 1, then the XNOR gate is selected because its output under these conditions is 1. The arrival of an edge on one of the MUT outputs propagates to the XOR or XNOR and generates the 1-to-0 transition of the negative pulse, and an edge (arriving later) on the second output generates the 0-to-1 transition of the pulse. The “EdgeCntA/B” components count the number of transitions that occur on each of the paths as a means of determining whether any glitching occurred. Although some types of glitching can be tolerated by the TDC, the relative timing value produced by the TDC can be corrupted by glitching and therefore we use only path tests in which the EdgeCnts are 1 for both paths. Paths for which this is true are called **stable paths**.

The difference in the delays of the two paths is captured in the width of the negative pulse. The TDC is designed to “pulse shrink” the negative pulse produced by the XOR/XNOR as it propagates down a current-starved inverter chain. As the pulse moves down the inverter chain, it activates a corresponding set of set-reset latches to record the passage of the pulse, where activation is defined as storing a 1. A thermometer code (**TC**), i.e., a sequence of 1s followed by a sequence of 0s, represents the digitized delay difference between the transitions occurring on the paths. A longer sequence of 1s (up to 120 in our version) reflects a wider initial negative pulse width.

A call-out of a current-starved inverter used in the delay chain is shown on the far right side of Fig. 13. The NFET transistor with input labeled “Cal_x” implements the current-starving mechanism. The “Cal_x” inputs are driven by two control voltages, labeled “Cal_0” and “Cal_1”. The current-starved inputs of all the even numbered inverters (numbered starting with 0) are connected to Cal_0 while the inputs of the odd numbered inverters are connected to Cal_1. This type of configuration allows independent control over the propagation speed of the two transitions associated with the negative pulse. For example, increasing the voltage on Cal_1 toward the supply voltage allows the odd numbered inverters to switch more quickly. With Cal_0 fixed at a specific voltage, larger

Cal_1 voltages allows the pulse to survive longer in the delay chain because it takes longer for the trailing edge to catch up to the leading edge. All latches up to the point where the pulse disappears store a 1, while those beyond that point store 0.

We determined that the highest resolution and lowest noise is achieved by setting Cal_0 to the supply voltage. Cal_1 is tuned for each chip in a calibration process described below. An off-chip voltage source drives the Cal_1 control signal in our experiments. Alternatively, this input signal can be derived from an on-chip resistor ladder network (a feature we intend to add to our next test chip). The TDC occupies an area of 176µm x 60µm (10kµm^2) at the 90nm technology.

4.2 TDC Sensitivity Analysis

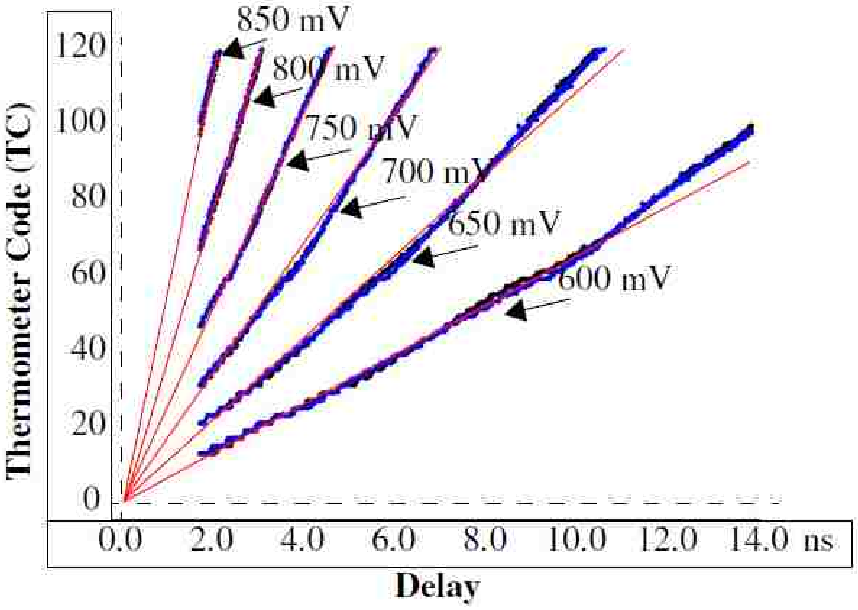

Figure 14: TDC sensitivity analysis: TC value produced by the TDC (y-axis) as the input pulse width is varied (x-axis) using a variety of Cal_1 values (separate curves).

The Cal_1 input allows the timing resolution to be tuned, trading off range and resolution. The curves in Fig. 14 illustrate the trade-off using data collected from one of the 90nm chips and one of the TDCs. The digital clock manager (DCM) from a Xilinx ZYNQ FPGA is used to drive a pulse into the CLK input shown on the left side of Fig. 13. The DCM allows the width of the pulse to be tuned with a timing resolution of approx. 18ps. The x-axis of Fig. 14 plots the pulse width which could be varied from approx 1.2ns (the smallest possible between the FPGA and the test chip board) up through approx. 14ns. The y-axis plots the TC produced by the TDC, a value between 0 and 120. The individual curves plot data using different values of Cal_1. For lower Cal_1 values, e.g., 600mV, the timing resolution is approx. 130ps per TC value while for higher values, e.g., 850mV, it increases to approx. 20ps per TC. The calibration process described above tunes Cal_1 to values between 750mV and 850mV across all chips, which provides approx. 25ps of timing resolution on average in our experiments.

4.3 HT Emulation Circuits

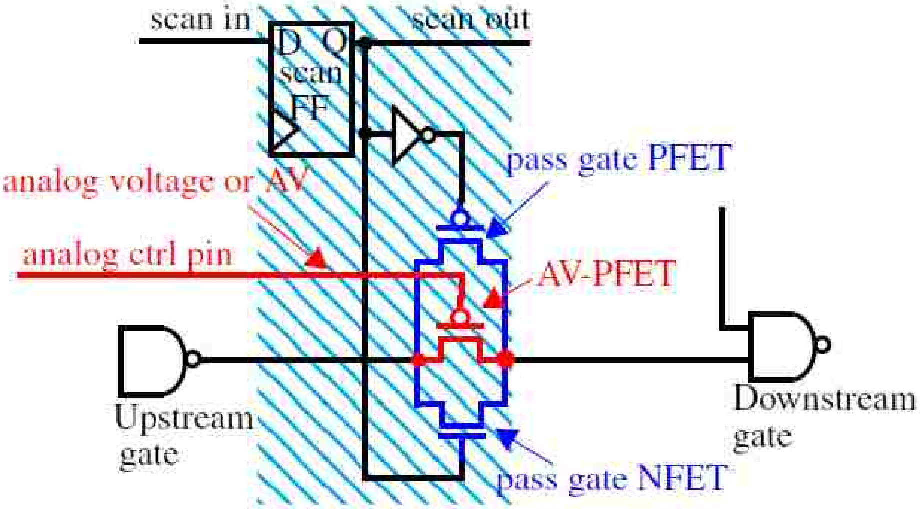

Figure 15: HT Emulation Circuit. Red Highlighted components are used to change the delay characteristics.

We inserted a set of 20 HT emulation circuits (HTEC) into each of the AES MUTs (the placement is identical in both AES units). A schematic representation of the HTEC is shown in Fig. 15. HTECs are inserted in series with an upstream and downstream gate at a variety of places in the MUTs netlists. It consists of a complementary NFET/PFET pass gate that can be enabled or disabled in a controlled fashion using a scan chain FF. An additional PFET, labeled AV-PFET, is connected in parallel with the pass gate, whose gate is controlled from an external voltage source using the analog control pin. With the pass gate disabled, the analog control pin can be used to change the resistance between the upstream gate and downstream gate, which, in turn, impacts the delay added by the HTEC. We refer to the applied voltage to the analog control pin as the analog voltage or AV. The HTEC cell is designed to allow the magnitude of the delay anomaly to be tuned, which in turn, is key to evaluating the capabilities and limitations of our proposed

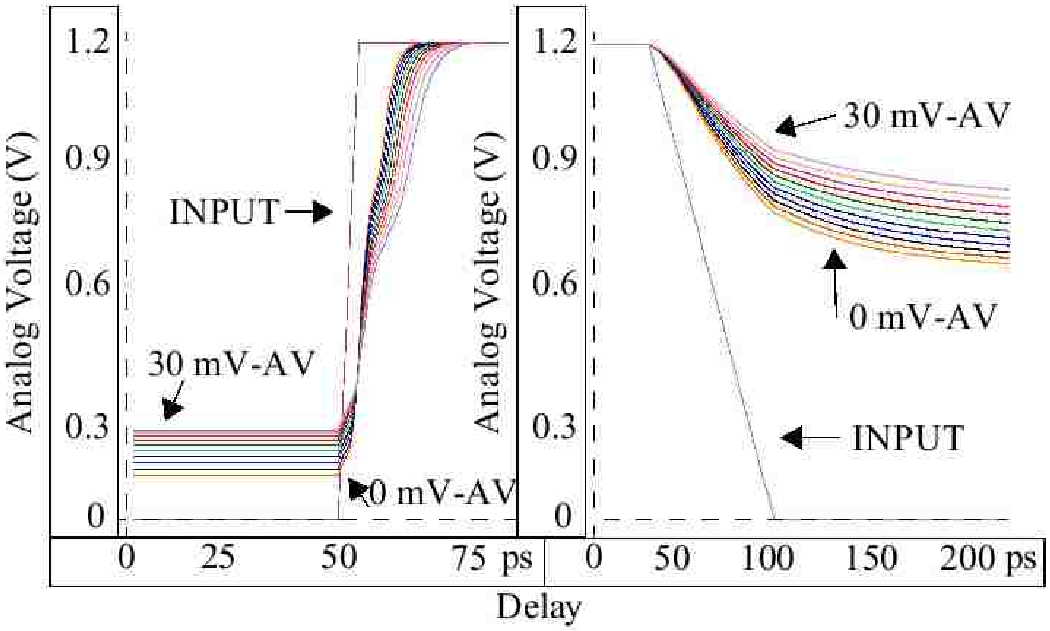

Figure 16: HTEC rise and fall time characterization. Rise-times vary from 35ps to 57ps. Fall-times vary from 288ps to 72,814ps. The AV varies from 0mV to 30mV with 3mV steps.

detection methods.

The HTEC delay properties are simulated to give understanding of delay contributions and variations. Fig. 16 displays the simulation-based rising and falling delay characterization that the HTEC provides with the AV varying from 0mV to 30mV by 3mV. The rising delay varies from 35ps to 57ps. The falling delay varies from 288ps to 72,814ps. This large variation in delay contributions delivers complex behavior especially in the HTECs that are deep within cones of logic.

Chapter 5
Accounting for Process Variations

5.1 Calibrating Chip-to-Chip Variations

The ability to tune the TDC using the Cal_1 input signal provides a mechanism to virtually eliminate chip-to-chip process variations effects. These effects include both a global shift and scaling of all path delays to values above or below the nominal value. We implement a calibration process that computes, for a fixed value of Cal_1, the average TC from a set of stable paths. A binary search process then tunes the Cal_1 value (and re-tests the paths) until the average TC equals a user specified value. We choose a value of 60 in our experiments, which is 1/2 the maximum TC value of 120. Note that calibration not only addresses chip-to-chip process variation effects but also reduces measurement bias in the on-chip TDCs.

5.2 Calibrating Within-Die Variations

Within-die variations currently represent one of the most significant challenges for parametric Trojan detection methods. We have developed a chip-averaging method that significantly reduces the adverse effects of within-die variations. Chip-averaging simply computes the average value of a path delay from measurements made from all chips (or a statistically significant sample). If within-die variations are random, then the averaging operation will eliminate them, revealing any type of systematic delay anomaly, i.e., an increase in delay that consistently occurs in all chips. The experimental results that we show in this paper illustrate that within-die variations are largely random and therefore, can be nearly eliminated using chip-averaging. This is a very powerful feature, and to the best of our knowledge, has not been previously proposed and/or demonstrated in hardware. Eq. 3 gives the expression for computing a chip-averaged-delay (CAD), where subscripts A,P indicate the AES unit (0 or 1) and the path ID (0 to n).

$$CAD_{A,P} = \frac{1}{Number\,of\,Chips} \sum_{1}^{Number\,of\,Chip} D_{A,P} \quad \textbf{Eq. 3}$$

5.3 Path Selection Criteria

Only paths that are classified as stable in both AES units across all 44 chips are used in this analysis. Also, a path is discarded from the analysis if the measured delay from any of the chips exceeds a 4σ limit. The 4σ limit for each path is derived by computing the standard deviation using the delays from all of the chips. Therefore, a delay that exceeds the limit is an outlier in the population. This catches paths that are unstable but are able to produce a single rising or falling edge and therefore are missed by the EdgeCnt check as discussed in reference to Fig. 3. The criteria for a path to be included in the analysis is very strict because we do not want glitching or small delay defects to bias the CAD value away from its nominal value, i.e., its value without process variations. A path coverage tool will be discussed later to address redundant data.

Chapter 6
Experimental Results

6.1 HT Emulation Circuit Experiments

A randomly selected sequence of 500 test vector pairs are applied to the two AES MUTs simultaneously. The same sequence is applied to each AES MUT to make it possible to directly compare the measured path delays. In these experiments, all HTECs pass gates are disabled as the vectors are applied, which means the pass gate controlled by the scan FF in Fig. 15 is in a high-impedance state and only the AV-PFET is conducting. Therefore, the delays of signals emerging on TDC inputs from paths sensitized from a HTEC site will depend on the value of AV. The same sequence of 500 vectors is applied 11 times to the 2 AES MUTs, each with a different AV applied. The values used for AV vary from 0.0V (referred to as the **HT-free** value since this value produces the smallest delay) to 0.3V in 0.03V steps. If a path is sensitized from a HTEC site, then the sequence of delays will change as the AV is varied. Note that the delays can increase or decrease depending on whether the edge effected by the HTEC cell is ahead or behind the edge on the HT-free path. We use this feature to distinguish between HT-free paths and HT paths.

Each TDC can be used to time 64 different combinations of the P_{Ax} and P_{Bx} inputs as each vector is applied. The 8 P_{Bx} inputs that are timed against TDC_x-A P_{A0} input, which is the CLK input, represent a special case because the arrival of the CLK signals remains constant for each test. Therefore, the length of the path driving the P_{Bx} input can be determined because it is proportional to the magnitude of the TC. The length of the paths under the other 56 combinations cannot be determined, however, but instead reflect the magnitude of the relative difference between the two edges.

Although it is possible to obtain up to 128 path delay measurements from the two TDC's for each MUT for each vector pair by repeatedly applying the test vector pair and selecting a different combination using the *Sel A* and *Sel B* MUXes, we ignore data obtained for TDC_x-B inputs 3 through 7. As mentioned earlier, we inserted extra gates in AES_2 to model actual HTs and these inputs are on sensitized paths from these inserted gates. A special set of experiments are carried out to determine the detectability of these hand-inserted HTs in Section 6.2. Therefore, only 24 of the 64 combinations, using P_{Bx} inputs 0, 1 and 2, in TDC_x-B are used in these experiments.

HT-free data is obtained from the AV = 0.0V data set, and for paths whose delay does not change with AV. A separate set of HT-free control data is also obtained from the data set with AV set to 0.0V but for paths whose delay does change as AV is increased (note: these are also HT-free paths when AV is 0.0V). The control data is used to determine how many false positive detections occur in our statistical detection method. The HT data is obtained using the HT-free control paths but with AV>0.0V. Given these constraints and those described earlier in Section 5.3, our post processing produced data sets with the following cardinality and characteristics:

- HT-free paths: 728
  - These are used to derive the statistical limits.
- HT-free control paths: 557
- HT paths: (557 * 2) per AV

- ○ Since each path that is affected by a HTEC occurs in both AES units, the number of HT paths that can be tested doubles. AVs between 0.03V and 0.30V in steps of 0.03V are used (for a total of 10 steps) in the HT experiments so 11140 HT paths are tested altogether.

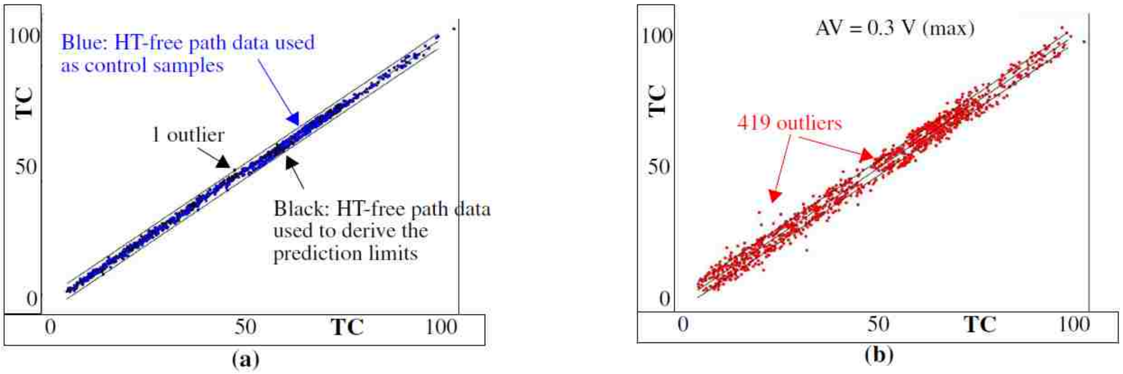

Figure 17: (a) Regression analysis of HT-free control samples from HTEC experiments, (b) using HT data points with AV=0.3V showing outliers.

Regression analysis is applied to the scatter-plot created using the 728 HT-free paths as shown in Fig. 17(a). Here, the CADs from paths in AES_1 are plotted against the CADs from the same paths in AES_2. The black data points represent the HT-free paths while the blue data points represent the HT-free control paths. A regression line and prediction limits are derived using the black data points. The limits were increased slightly above 3σ (99.73) to 3.5σ (99.9) to account for the slight increase in noise that occurs when we use path delays from all combinations of TDC inputs (as opposed to paths that are timed against the CLK only). Only one outlier is present in the data as highlighted in the figure (Fig. 17) (from the HT-free data). All HT-free control data points fall within the limits. The average width of the band is approx. +/- 2.5 TCs, which means that uncompensated process variations and noise add uncertainty of approx. 63ps.

Fig. 17(b) plots the HT data points with AV=0.3V (the largest value that we used in our experiments) on top of the prediction limits to illustrate the number of outliers for this scenario. The data point pairings in this case are created by using the AV=0.0V CADs from AES_1 as the first element of the pairing and the AV=0.3V CADs from AES_2 as the second element (and vise-verse). In other words, the data point pairings are created from a combination of HT-free and HT CAD values. This is consistent with the self-referencing technique where it is assumed that the adversary only adds the HT to one of the functional units.

Table 2 shows the detection results, partitioned into rows according to the applied AV. The 1st column gives the AV value, the 2nd column gives the average increase in the TC (and the corresponding delay), while the last column gives the number of detections (and as a percentage of the total). The average increase in TC is computed using the differences in the HT-free control CAD's (with AV=0.0V) and the HT's CADs (with AV>0.0V) across all 1114 paths. As indicated earlier, 1 TC ~= 25ps. Although some of the 40 HTECs (20 in each AES) are detected for lower values of AV, detecting them all is not possible until the delay anomaly becomes greater than 1 TC. A 5X increase in the level of confidence of a positive detection occurs when the anomaly reaches 2 TCs. For perspective, the fanout of 4 gate delay is approx. 35ps for a 1X inverter in this technology. Therefore, these results indicate that it is possible to detect an extra inverter in a path with reasonably high confidence.

AV (V)	Average increase in TC (ps) over HT-free path	#(%) of paths that detect an HT (out of 1114)
0.03	0.27 (7ps)	3 (0.3%)
0.06	0.53 (13ps)	7 (0.6%)
0.09	0.79 (20ps)	34 (3.1%)
0.12	1.05 (26ps)	67 (6.0%)
0.15	1.30 (33ps)	115 (10.3%)
0.18	1.55 (39ps)	176 (15.8%)
0.21	1.79 (45ps)	246 (22.1%)
0.24	2.02 (51ps)	302 (27.1%)
0.27	2.25 (56ps)	359 (32.2%)
0.30	2.46 (62ps)	419 (37.6%)

Table 2: HT detection results from HTEC experiments.

6.2 Fan Out and Series Inserted HT Experiments

The objective of these experiments is to determine if it is possible to detect a set of HTs implemented with standard cell logic gates connected as fan-out loads or in series with paths in the AES$_2$ MUT. Similar to the self-similarity strategy taken in the previous section, we compare path delays from two “nearly” identical copies of the AES to determine if systematic anomalies can be measured and detected on paths that posses an HT.

6.2.1 Fan Out and Series Inserted Trojans

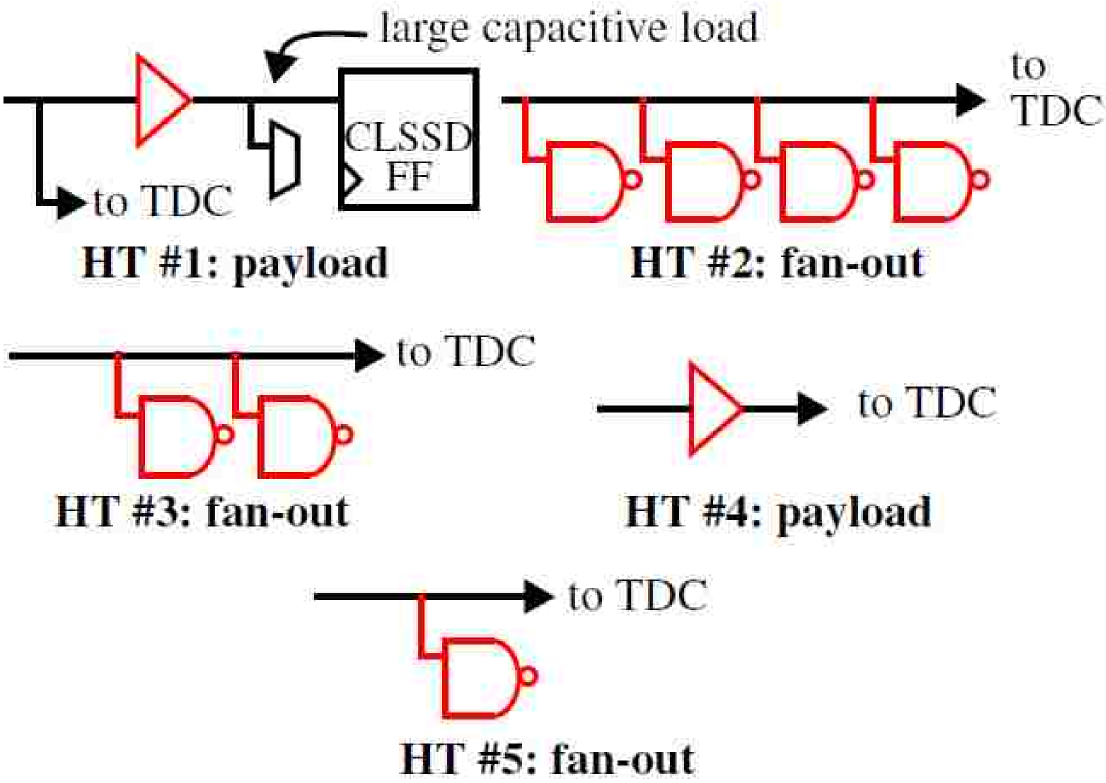

Figure 18: Manually inserted fan-out and payload HTs in AES$_2$.

As mentioned earlier, we purposely made the layouts for AES$_1$ and AES$_2$ (including the TDCs) nearly identical. The Cadence synthesis and place and route tools were used to create one copy of the layout that was then placed into the two locations as shown in Fig. 11. The layout editor was then used to make small changes in the copy labeled AES$_2$. In particular, filler cells were removed and additional gates were added as shown in Fig. 18. The red components represent the HT while the black components represent the existing elements in the layout. Each HT was designed to effect only one of the AES outputs. HT #1 adds a buffer which isolates the large capacitive load present on the capture FF input from the wire that connects to the TDC input. Therefore, the delay along paths to the output for AES_2 should be less than the delay for AES_1 which does not include the isolation buffer. HTs #2, #3 and #5 add capacitive load to existing wires while HT #4 includes a series inserted buffer. Therefore, the path delays from AES_2 for HTs 2 through 5 are expected to be longer than those measured from AES_1.

6.2.2 Methodology

We again applied a set of 500 random vector pairs to the 2 AES MUTs. Unlike the analysis carried out in the previous section, all TDC path measurements are carried out using the CLK as reference, i.e., only P_{A0} from Fig. 12 is used, and all HTECs are disabled. The HTs referred to in Fig. 18 affect 5 of the TDC_2-B P_{Bx} inputs while the remaining 11 TDC inputs (8 from TDC_2-A and 3 from TDC_2-B) are HT-free.

6.2.3 Regression Detection Results

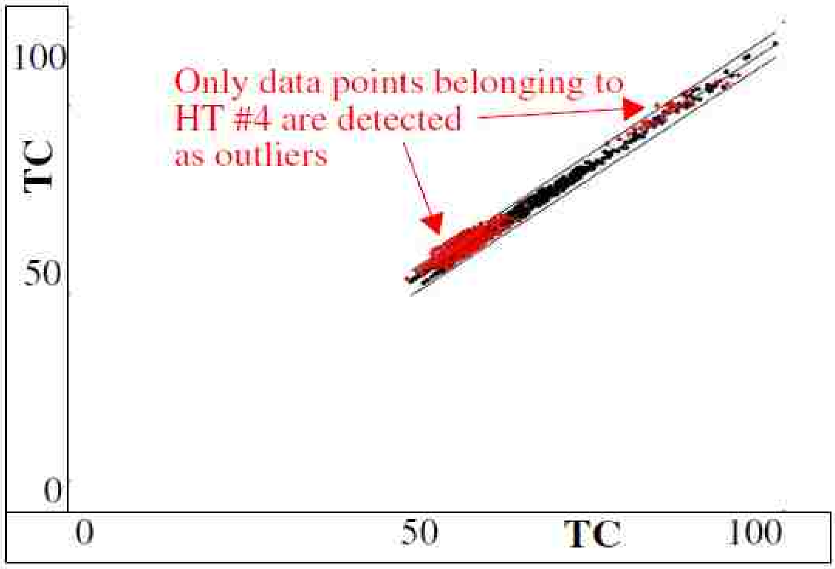

Figure 19: Regression analysis of fan-out/payload HTs.

Fig. 19 plots the CADs for AES_1 along the x-axis against the CADs for AES_2 along the y-axis (similar to the plots described in reference to Fig. 17). A regression line and 3σ prediction limits are derived using 676 HT-free CADs, while the HT CADs for 468 paths are shown in red. Path delays begin at approx. TC=50 because the CLK edge always arrives on the TDC P_{A0} before any of the transitions propagating through the AES MUTs arrive on the TDC P_{Bx} inputs. Outliers occur in the region as labeled. Unfortunately, only HT #4 is detected using regression analysis. Therefore, regression is not sensitive enough to detect most of these subtle HT delay anomalies.

6.2.4 Trending Detection Results

In this section, we develop an alternative detection method that is based on the analysis of trends in the CAD differences. The trends introduced by systematic anomalies are captured by computing a **difference** CAD or DCAD value defined by Eq. 4.

$$DCAD_{A,P} = CAD_{1,P} - CAD_{0,P} \quad \textbf{Eq. 4.}$$

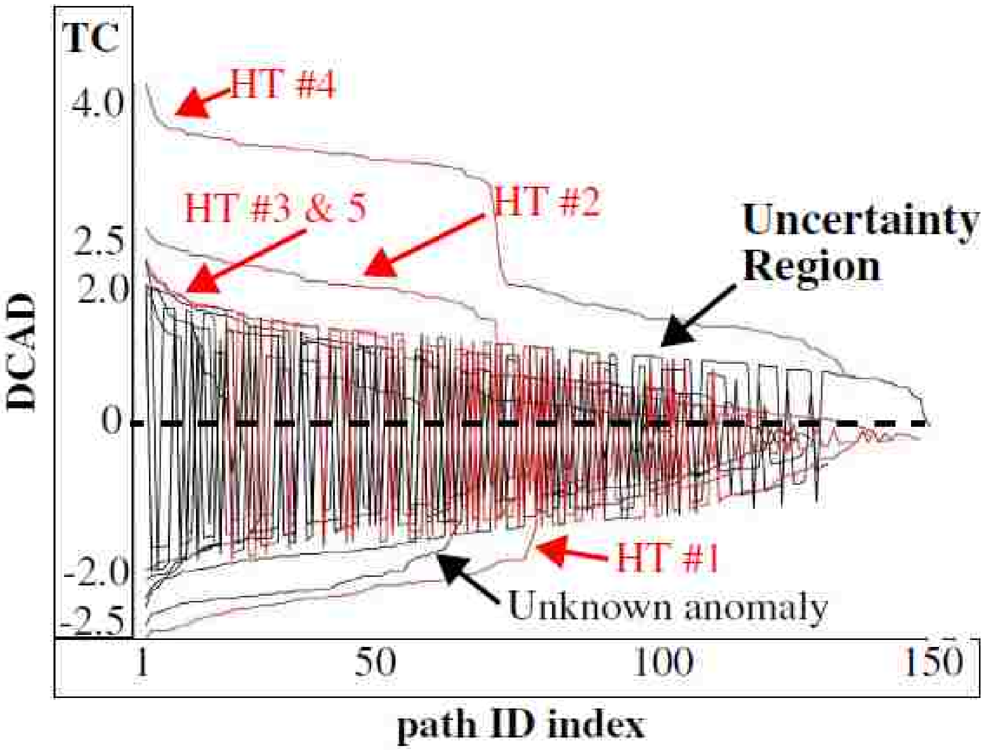

Figure 20: DCAD values on y-axis for paths tested through each of the TDC inputs, sorted on the magnitude of the differences from largest (left) to smallest (right).

We partition the DCAD values computed for the stable paths under all 500 vectors into 16 groups, with each group corresponding to one of the HT or HT-free TDC inputs. Each of the groups of DCAD values are plotted as a curve in Figs. 20 and 21. All 16 curves are superimposed in Fig. 20 while Fig. 21 partitions the curves into HT and HT-free groups. The data points in each curve represent all of the stable paths that were tested. We only include paths that go through the HT site for the HT TDC inputs, and ensure that all paths included in the HT-free curves are HT-free.

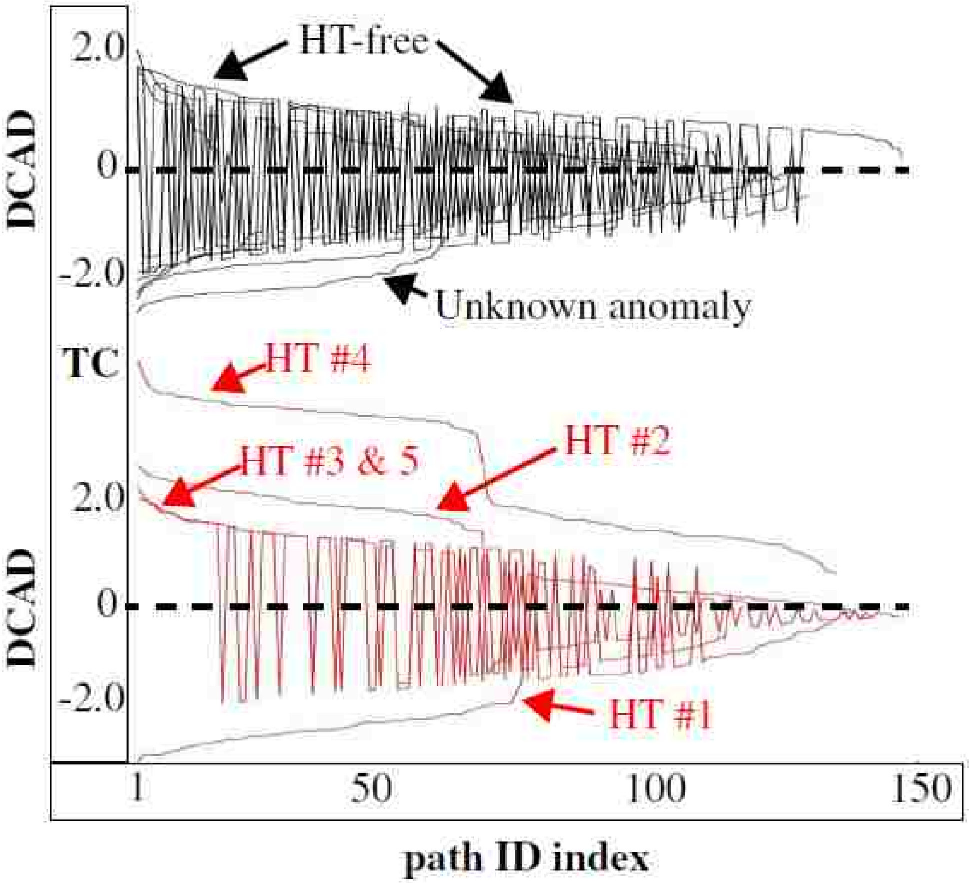

Figure 21: Same as Fig. 19 except HT-Free and HT results are separated.

The values are sorted from left-to-right on the magnitude of the DCAD value, with the largest differences on the left. The shaded region labeled "Uncertainty Region" represents the noise floor, i.e., the point at which the DCAD values in the curves begin jumping between negative and positive values in a random fashion. The cone-like shape, with a maximum width of +/- 2 TCs on the left, indicates that noise is proportional to the magnitude of the difference. The ordering of the data points along the x-axis also correlates to path length, with the data points on the left side of the curves produced by the longest paths.

The curves above and below the Uncertainty Region capture a systematic component of variation in path delay, i.e., a shift in delay that is similar in all chips. The 5 HT curves are labeled, and are consistent with the expected behavior as described in Section 6.2.1. For example, HT #1 isolates the large load capacitance of the CLSSD capture FF and therefore, the delay in AES_2 is smaller than the delay in the AES_1 (the nominal version), producing a curve with negative values. Larger positive increases in delay are expected in the curves for HT #2 and #4, while the expected increase in delay of HT #3 and #5 should be smaller. The average TC over the left portion of the curves is 2.5, 3.75, 2.0 and 2.0 for these 4 HTs, which is consistent with the expectations. Note that the curves for HT #3 and #5 are very close to the noise floor, making it difficult to decide with high confidence that a systematic anomaly exists. Therefore, fan-out loads of 1 or 2 gates represent the most challenging HT characteristics to detect. However, the trend of the data points for the HT inputs to remain for at least some fixed interval above or below the 0 line provides evidence of a systematic anomaly. Notice that this representation, unlike the regression curve in Fig. 19, clearly shows that HT #1 and #2 are anomalous.

All 11 of the HT-free curves in Figs. 20 and 21 fall within the Uncertainty Region as expected except for 1 curve. The reason that a systematic anomaly exists in the curve labeled “Unknown anomaly” is unclear. We carefully checked the layout for wiring differences between the 2 AES units for this TDC input and found only small differences in capacitance coupling introduced by cross-over wires that are present in AES_2 but not in AES_1. We do not believe these are significant enough to introduce the observed anomaly. We also tested each of the 4 TDCs across all chips using a series of clock pulses (similar to that described in reference to Fig. 14), computed chip-averages and compared the CAD values of TDCs connected to AES_1 with those computed for AES_2. The differences were less than 1 TC, which indicates that the TDCs themselves are not introducing the anomaly. The fact that the anomaly is present, and is consistent across a large number of paths and chips makes it likely that there is a physical difference in the layouts (which we cannot find), or there is some type of systematic process variation (which is unlikely because this would affect more than just one of the TDC inputs).

6.2.5 Analysis of the Effectiveness of Chip-Averaging

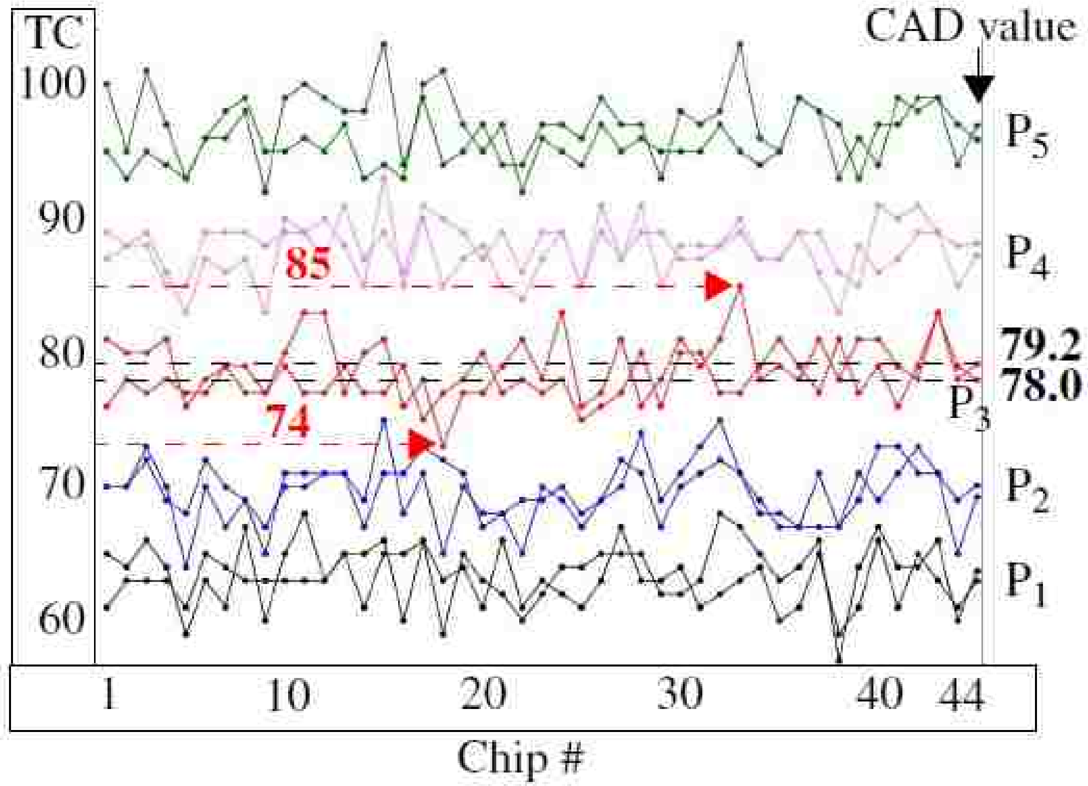

Figure 22: TCs from individual chips and the chip-averaging delay (CAD).

In this section, we quantify the level of reduction in noise and within-die variations provided by chip-averaging. As indicated above, the CAD values are computed from the TC values measured from 44 chips. Fig. 22 plots the individual chip TCs for a set of 5 HT-free paths, labeled P_1 through P_5, as the first 44 values along the x-axis and the CAD value as the last (right-most) data point. The two curves associated with each path are derived from each of the two AES units. The length of the path increases from P_1 through P_5 as reflected in the range of TCs. The variation in the individual chip values is caused by noise and within-die process variations. Given all 5 paths are HT-free, the CAD values for each pairing of paths on the far right should be equal in the absence of noise. The difference between the CAD values, defined earlier as DCAD, in all 5 cases is close to the ideal case of 0. The largest DCAD value is 1.2, as given by P_3 CAD values of 79.2 and 78.0. The range of variation in the individual chips, on the other hand, is given by 85 - 74 = 11. Therefore, chip-averaging reduces undesirable variations introduced by noise and within-die variations by almost a factor of 10, allowing small systematic variations introduced by HTs to be more easily detected.

6.3 Trojan Detection Conclusions

We present chip results of using a self-similarity based method to detect hardware Trojans (HTs). An embedded test structure called a time-to-digital converter is used to obtain high resolution, e.g., 25ps, measurements of path delays from two nearly identical copies of AES. A chip-averaging technique is shown to significantly reduce the adverse effects of within-die process variations on HT detection sensitivity. Statistical methods are used to detect path delay anomalies introduced by HT emulation circuits and HTs implemented by the insertion of standard cell logic gates.

Chapter 7
Path Coverage and Simulation Analysis

7.1 Path Coverage

The analysis of the hardware was simplistic and did not account for the intrinsic nature of the circuit. We used 500 random vectors and hoped that we tested unique paths and did not retest paths a large amount. To verify, we first spent a significant amount of time attempting to have the Encounter Test tool from Cadence yield our answer. After some time with the tool, the need to develop a structural path analysis tool arose. The tool took a structural netlist and a waveform as input. The tool mapped all structural paths into a set of data structures that with the waveform data would give structural paths that any transition followed. We used Cadence's NCSim to create simulations for all 500 vectors. These simulations leveraged the delay information that was abstracted from the standard cell implementation of the AES_1 macro. We passed these waveforms to the tool and were able to remove repeated paths from our datasets. The analysis returned that each of the HTs were tested uniquely between 270 to 300 times in the hardware. With this information, we could target specific vector pairs that yield the most transitions on HT outputs. The Fig. 23 and Fig. 24 show the results of removing redundant data from the counterpart figures earlier Fig. 20 and Fig. 21. The figures display the importance of this analysis showing the fact that the anomaly discussed prior all but disappears when we removed the redundant data.

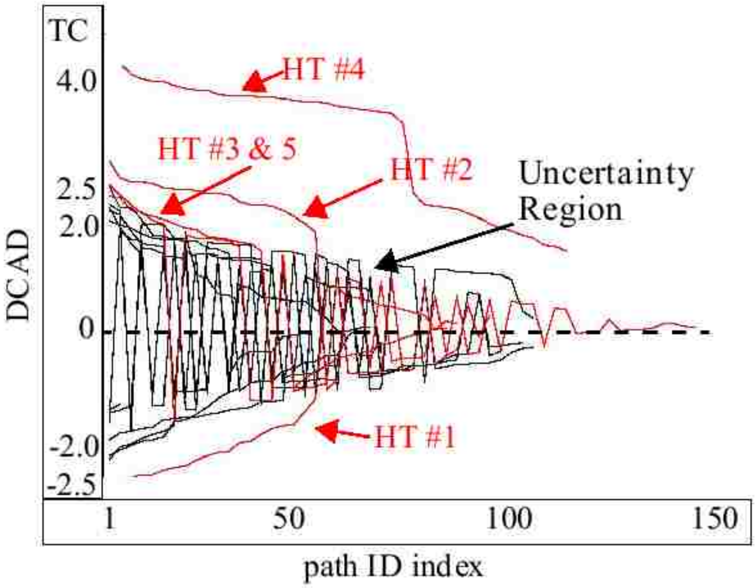

Figure 23: DCAD values (CAD(AES2)-CAD(AES1)) on y-axis for unique paths tested through each of the TDC inputs, sorted on the magnitude of the differences from largest (left) to smallest (right).

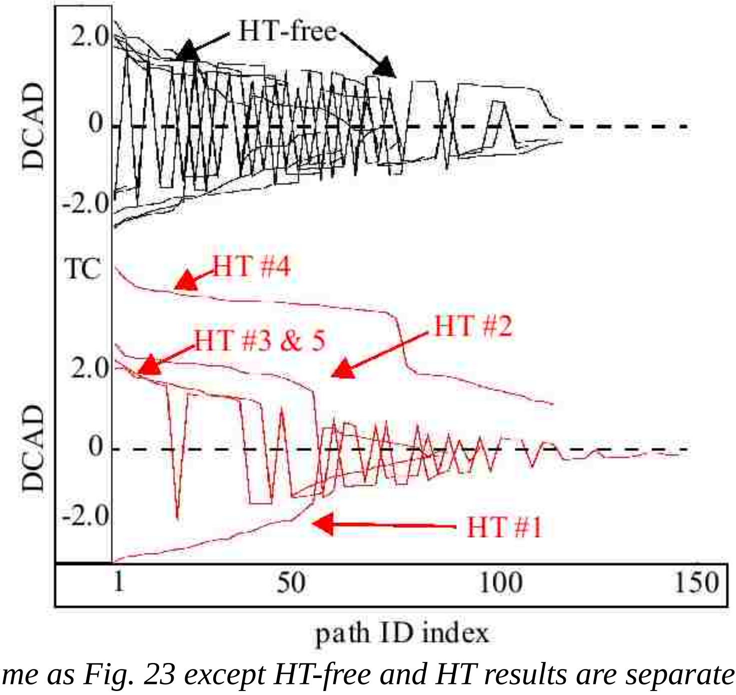

Figure 24: Same as Fig. 23 except HT-free and HT results are separated.

The path coverage analysis allowed with more certainty to detect HTs 1, 2 and 4 while better illustrating that HTs 3 and 5 are on the edge of being detectable. Later, we will address detectability when using the simulation model as the golden-model.

There is also a parallel analysis presented by doing the path coverage statistics, we sought to investigate if there exists a relationship between the magnitudes of the DCADs and the path lengths, specifically of the HT paths. Since we were able to have structural paths that we analyzed over all vectors per transition, we could make Fig. 25 and see if there was anything that could be addressed.

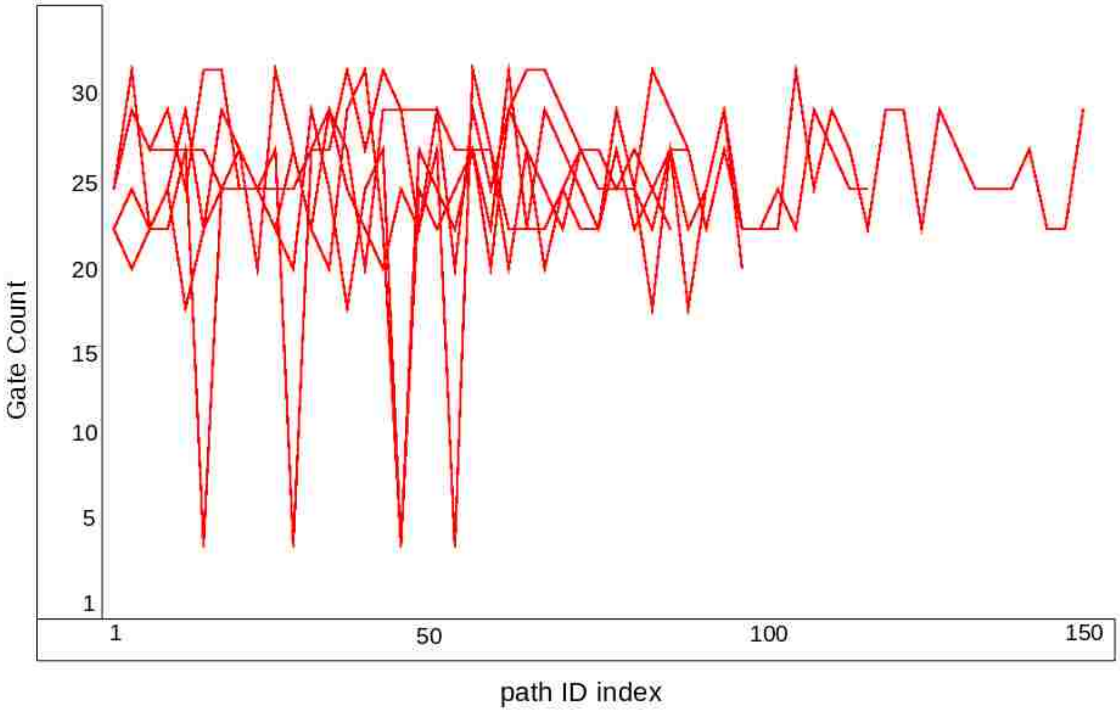

Figure 25: Path ID index takes the same order as Fig. 23 where the IDs are sorted by magnitude of their respective DCADs. Path length represented by gate counts are represented on the y-axis. The separate curves are of the separate HTs.

From the synthesis of the macro, there are no paths of length 4 to 15 gates. This fact makes deduction of a trend in detectability while accounting for path length difficult. We expected that the shortest paths were the easiest to detect. Thus we thought this Fig. 25

would have a trend going from the bottom left to the top right. Instead, we see that generally our few short paths are on the left but at the same time there are paths of varying lengths also on the left. Therefore we cannot say that shorter paths are more detectable. With a different macro where there is an even distribution of path lengths, we may be able to conclude some more concrete findings.

7.2 Macro Simulation

The simulations designed to complement the hardware experiments were broken into the AES_1 macro and TDC parts. AES_2 was not simulated because it contained the inserted gates and we are trying to build the simulation-based golden model. This separation was necessary considering that with a single simulation of the AES_1 we will obtain as many data points as there are transitions on outputs. If the TDC were entangled with the AES ones, there would require single simulation for each and every transition on an output. Instead, we ran simulations of the TDC at a representative calibration voltage with varying pulse widths. We then used the TC's generated paired with each corresponding pulse-width to translate analog delay values from the AES_1 to compare with the hardware results. We used Mentor Graphics Calibre XRC extractor to generate RC or RCC models as needed. We simulated these models with Cadence Spectre transient solver ran for a set simulation interval where after all circuit activity had completed. Nodesets were leveraged to set the scan-chains before simulation started to the desired vector values.

7.2.1 AES Simulation

The AES macro was extracted into both RC and RCC models and the data was analyzed. The time difference between the two simulations was significant enough (in the range of RC taking ~3-5 hours and RCC taking about ~173 hours) that we decided that the small change in the data from adding coupling capacitance to the model was not worth the effort. We ran 20 of the 500 vectors. These 20 vectors were hand chosen based upon the path coverage analysis to generally have more unique output activity than other vectors. Although a simple transistor and R based models were tested for timing length of simulation, the data was not meaningful without at least the base capacitors in the routing wires. The simple transistor model took only 8 minutes with the matured simulation flow. The length of the R model simulation was not accounted for with the matured simulation flow and thus the reported time for the R model was 40 hours. The biggest speed that was achieved was by setting the DC solver to a known algorithm that was generally successful. Most of the 40 hours spent working on the R model was in trying multiple DC solvers, after which, the tool decided to use the solver that we set to be used consistently after.

With the extraction set, delay waveforms were made per output of the AES. Filtering occurred to remove noisy, glitchy, and redundant data from the output waveforms. We can use these waveforms to get the pulse-width from the edge relative to the clock pulse timings. Once we had the pulse-widths, we can look up the corresponding TC from the TDC simulation described below.

7.2.2 TDC Simulation

The TDC simulation was known to need high precision and thus an RCC model was used from the start. Even with the RCC model, the TDC model was a tenth the size of the AES-RC model. Therefore, we could many simulations or in this case a single, long simulation in a timely fashion. We set out to get pulse-widths translated into TCs at representative calibration voltages. We ran 11 pulse-widths ranging from 500ps to 2100ps at calibration voltages from 700mV to 850mV. This process was similar to that of the TDC characterization done in hardware but with precise inputs. The hardware saw effects that put noise on the inputs and thus precise relation of input pulse-width to TC was somewhat fuzzy. We were able to run this simulation in a one-off circumstance which took a little over a day. With this simulation ran, we extracted the waveforms of each flip-flop in the TDCs delay chain and correlated those waveforms to TCs. Specifically, when we saw a transition on one the ordered flip-flops we incremented a counter. After having TCs for 11 pulse-widths, these data values were read into a MATLab script that interpolated points in between with a resolution of 0.1ps. We would then be able to round any delay from the AES block to a tenth of a picosecond and have a digitized value for the analog delay developing a digitization process akin to a look-up table. To choose the effective calibration voltage, we took the average of all scaled calibrations used in the hardware. We intended this to be a nominal-like calibration situation. Some noise or error in our simulation analysis could be attributed to this forced calibration. In the end, we chose this avenue because the calibration should be a simple shift of the data and is leveraged to prevent overflow. In the simulation realm, we should never see overflow because the AES block is nominal and does not vary.

7.2.3 Simulation Results

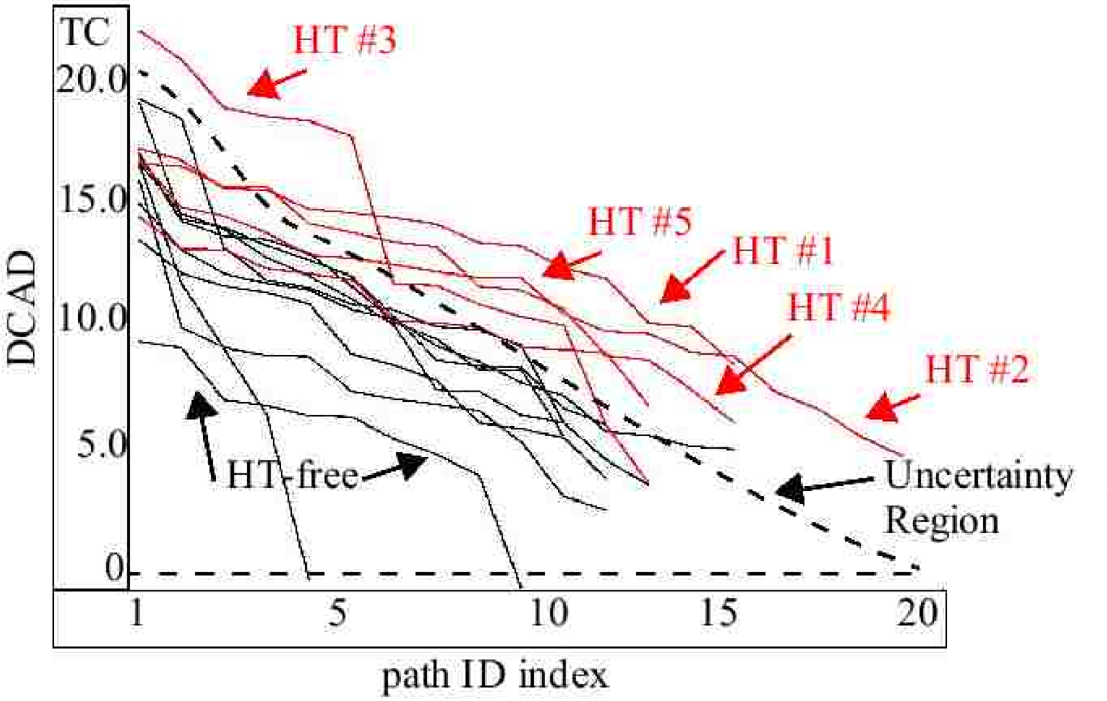

Figure 26: Path ID index takes the same order as Fig. 23 where the IDs are sorted by magnitude of their respective DCADs. DCADs taken from the simulation golden-model differenced with the AES$_2$ are on the y-axis.

Fig. 26 shows analogous results to Fig. 23. We took the DCAD of the simulation data and the AES$_2$ hardware results and plotted the values on the y-axis (CAD(AES$_2$)-SIM). The x-axis is again, the path ID sorted by DCAD magnitude. The analysis clearly indicates that the HTs are still detectable. There is far less intuition to infer the order of the HTs. With the shift of the data caused by the simulation being generally significantly slower than hardware, the scale also gets obscured. Although, the HTs are not ordered as expected, they are still generally grouped together above the “Uncertainty Region” and therefore are detectable. The lack of the conical behavior can be attributed to the overall shift of the simulation data caused by hardware-to-model correlation as well as other implicit effects. Within the “Uncertainty Region”, we still could not determine HTs from HT-free data. The performance of the simulation-based

DCAD analysis sheds light onto the larger issue of using hardware-to-hardware based comparisons. After doing rigorous statistical limiting and averaging, the AES_1 data still was significantly different from the simulation-based golden model. The results from the simulation-based DCAD analysis gave motivation to analyze the complicated nature of why the hardware was still significantly variant to the simulation golden-model. The chip we designed had a heterogeneous consistency, meaning that varying parts of the die looked significantly different. The implicit effects of being printed on a larger die with this consistency can be categorized into neighborhood effects, routing effects, and more subtle systematic effects that are unaccounted for. These implicit effects also attempt to explain what subtle differences could exist between these two seemingly nearly identical macros. The neighborhood effects in the design are caused by inconsistencies in power grid, macro placement, and position relative to the edge of the chip. AES_1 has a large macro to the east of itself while AES_2 has mostly open space, as evident in Fig. 11. This difference would cause large edge effects that would differ from macro to macro. The power grid would also see variation caused by different leakage and switching profiles of the neighboring areas. Also within the chip layout, the relative placement of them across to the edge of the chip, could create varying edge effects. Although the macros core logic is exactly the same, several of the routing channel differ slightly. After place and route occurred in the AES_1, the AES_2 was a direct copy and paste of the previous. The routes that are slightly different would be the clock tree, which should be balanced but does contain variation, and the external routes to other resources, the TDC for example. The routes to the TDC were made with extensive effort to match one another but there are small variations from one AES to the other, especially taking into consideration the placement of the macros relative to the TDCs on the chip. Both of these routing artifacts would present shifts in the data that chip-averaging would not be able to compensate for. Lastly, as with all chip manufacturing there always exists systematic changes caused by process variations. We have spent effort with extensive statistical analysis to attempt to compensate and correct for the major systematic effects but more subtle ones could be contributing to the discrepancies. These systematic effects are typically from one part of the die to the other. The over-all size of the macros make the subject to these effects. The degree of systematic effects are difficult to judge. Chip-averaging and outlier restrictions should be able to clean up some of the systematic effects but there are many artifacts that could still be in the data.

Chapter 8
Future Work and Conclusions

8.1 Future Work

The research that has been completed to this point has been two-fold; PU's and Trojan Detection. The PUF research looked into an IP-Level implementation of a metal resistance based encryption and authentication system. The Trojan detection looked into using an embedded test structure (ETS) in the TDC to digitized delays that could be statistically analyzed to assess the presence of Trojans.

The marriage between the two works would be to utilize the TDC as an ETS in a PUF. In this scenario, the TDC could be used as a Trojan detector with peripheral hardware as needed at manufacturing time and a digitizer in the field of entropy sources distilled from working circuits.

Specifically within the realm of hardware Trojan detection, there is work to be done with new hardware at an untested technology node. We could design a new form of TDC where the delay chain is distributed among the chip giving the ability to measure an even larger scope of path lengths.[42] The implementation of a larger scope of HT models, i.e. implementing realistic Trojans that actually do work. There is an argument to refine the design and fabricate it at the same technology node to compare results. There is also another argument to implement in a new node and account on the variations between the data we have and the new data that we would get. With testing rigor, we have also decided that the HTEC that we have built could be implemented much better to emulate various types of Trojans.

The development of the path coverage tool could open opportunity to enable IP-Level detection of hardware Trojans. Significantly more software development would be required to create this Design-for-Security tool as a security checker of sorts, but the foundations have been laid in this tool.

The testing algorithm to simulate the TDC with the AES could be improved. There is significant noise and shift of the data caused by not correctly calibrating the data. There could be work exercised in developing a sort of calibration algorithm for the simulation platform.

8.2 Conclusions

In this thesis, hardware security areas are discussed. Specifically, a Resistance-based IP-Level PUF system is described in a 65nm technology with statistical metrics provided. The work continued with a look into hardware Trojan detection using an embedded test structure (ETS) known as a time-to-digital converter (TDC). The TDC has been used in a 90nm technology to digitize path delays of an implemented AES Macro-under-test. Using chip averaging and a trending analysis, inserted gates can be detected based upon a Trojan-free versus Trojan-inserted comparison of the MUT. The research has been successful in providing results to this point and shows promise in providing improvements in both areas of hardware security.

References

[1] "Secure, Trustworthy, Assured and Resilient Semiconductors and Systems (STARSS) Workshop", sponsored by SRC/NSF, San Jose, CA, May 21, 2014, https://www.src.org/cal-endar/e005440/

[2] "Design for Security Working Meeting", sponsored by USC,ISI and US Army Research Office, Marina del Rey, CA, July 23, 2014, https://uscisi.atlassian.net/wiki/display/DFSWM

[3] J. Ju, R. Chakraborty, C. Lamech and J. Plusquellic, "Stability Analysis of a Physical Unclonable Function based on Metal Resistance Variations", HOST, 2013, pp. 143-150.

[4] R. Chakraborty, C. Lamech, D. Acharyya and J. Plusquellic,"A Transmission Gate Physical Unclonable Function and On-Chip Voltage-to-Digital Conversion Technique", DAC, 2013,pp. 1-10.

[5] B. Gassend, et al., "Controlled Physical Random Functions,"Conference on Computer Security Applications, 2002.

[6] K. Lofstrom, et al., "IC Identification Circuits using Device Mismatch," SSCC, 2000, pp. 372-373.

[7] M. Kalyanaraman and M. Orshansky, "Novel Strong PUF-Based on Non-linearity of MOSFET Subthreshold Operation",HOST, 2013, pp. 13-18.

[8] M. Majzoobi, et al., "Lightweight Secure PUFs", ICCAD, 2008.

[9] A. Maiti and P.Schaumont, "Improving the Quality of a Physical Unclonable Function using Configurable Ring Oscillators", FPLA, 2009. pp. 703-707.

[10] Y. Meng-Day, et al., "Performance Metrics and Empirical Results of a PUF Cryptographic Key Generation ASIC," HOST,2012, pp. 108-115.

[11] C. Yin, G. Qu, and Q. Zhou, "Design and Implementation of a Group-Based RO PUF", DATE, 2013, pp. 416-421.

[12] J. Guajardo, et al., "Physical Unclonable Functions and Public Key Crypto for FPGA IP Protection," FPLA, 2007, 189-195.

[13] S. Okumura, S. Yoshimoto, H. Kawaguchi,and M. Yoshimoto,"A Physical Unclonable Function Chip Exploiting Load Transistors' Variation in SRAM Bitcells", ASP-DAC, 2013, pp.22-25.

[14] Y. Alkabani, et al., "Trusted Integrated Circuits: A Non-destructive Hidden Characteristics Extraction Approach," Information Hiding, 2008.

[15] Y. Yao, M. Kim, J. Li, I. Markov, and F. Koushanfar, "Clock-PUF: Physical Unclonable Functions Based on Clock Networks", DATE, 2013, pp. 18-22.

[16] Z. Yu, A. Krishna, and S. Bhunia, "ScanPUF: Robust Ultralow-Overhead PUF Using Scan Chain", ASP-DAC, 2013, pp.626-631.

[17] K. Cho, K. Lee, S. Kim, S. Lee, and Y. You, "Implementation of a Physical Unclonable Function (PUF) with Transmission Line Crosstalk in a Chip", ASQED, 2013, pp. 240-244.

[18] NIST: Computer Security Division, Statistical Tests, http://cs-rc.nist.gov/groups/ST/toolkit/rng/stats_tests.html

[19] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, B. Sunar,"Trojan Detection using IC Fingerprinting", Symposium on Security and Privacy, 2007, pp. 296 -

310.

[20] F. Wolff, C. Papachristou, S. Bhunia, and R. Chakraborty, "Towards Trojan-Free Trusted ICs: Problem Analysis and Detection Scheme", DATE, 2008, pp. 1362-1365.

[21] R. S. Chakraborty, S. Paul and S. Bhunia, "On-Demand Transparency for Improving Hardware Trojan Detectability", HOST, 2008, pp. 48-50.

[22] M. Banga and M. S. Hsiao, "ODETTE: A Non-Scan Design-for-Test Methodology for Trojan Detection in IC's," HOST,2011, pp. 18-23.

[23] D. McIntyre, F. Wolff, C. Papachristou, S. Bhunia and D.Weyer, "Dynamic Evaluation of Hardware Trust," HOST, 2009, pp. 108-111.

[24] H. Salmani and M. Tehranipoor, "Layout-Aware Switching Activity Localization to Enhance Hardware Trojan Detection", Trans. on Information Forensics and Security, Vol. 7, Issue: 1, Part: 1, 2012, pp. 76-87.

[25] M. Banga and M. S. Hsiao, "A Region Based Approach for the Identification of Hardware Trojans", HOST, 2008, pp. 40-47.

[26] J. Zhang, Y. Haile and X. Qiang, "HTOutlier: Hardware Trojan Detection with Side-Channel Signature Outlier Identification", HOST, 2012, pp. 55-58.

[27] L. Wang, H. Xie and H. Luo, "A Novel Analysis Method of Power Signals for Integrated Circuits Trojan Detection", International Symposium on Physical and Failure Analysis of Integrated Circuits, 2013, pp. 637-640.

[28] C. Lamech, R. M. Rad, M. Tehranipoor, J. Plusquellic, "An Experimental Analysis of Power and Delay Signal-to-Noise Requirements for Detecting Trojans

and Methods for Achieving the Required Detection Sensitivities," Transactions on Information Forensics and Security, Vol. 6, No. 3,2011, pp. 1170-1179.

[29] M. Potkonjak, A. Nahapetian, M. Nelson and T. Massey,"Hardware Trojan Horse Detection using Gate-Level Characterization," DAC, 2009, pp. 688-693.

[30] D. Rai and J. Lach, "Performance of Delay-based Trojan Detection Techniques under Parameter Variations", HOST2009, pp. 58-65.

[31] C. Lamech and J. Plusquellic, "Trojan Detection Based on Delay Variations Measured using a High-Precision, Low Overhead Embedded Test Structure," HOST, 2012, pp. 75-82.

[32] A. Davoodi, L. Min and M. Tehranipoor, "A Sensor-Assisted Self-Authentication Framework for Hardware Trojan Detection", IEEE Design & Test, Vol. 30, Issue: 5, 2013, pp. 74-82.

[33] I. Exurville, J. Fournier, J.-M. Dutertre, B. Robisson and A.Tria, "Practical Measurements of Data Path Delays for IP Authentication & Integrity Verification", Workshop on Re-configurable and Communication Centric Systems-on-Chip, 2013, pp. 1-6.

[34] W. Sheng and M. Potkonjak, "Malicious Circuitry Detection using Fast Timing Characterization via Test Points", HOST,2013, pp. 113-118.

[35] C. Byeongju and S. K. Gupta, "Efficient Trojan Detection via Calibration of Process Variations", Asian Test Symposium, 2012, pp. 355-361.

[36] Y. Jin and Y. Makris, "Hardware Trojan Detection Using Path Delay Fingerprints", HOST, 2008, pp. 51-57.

[37] Jie Li and John Lach, "At-Speed Delay Characterization for IC Authentication and Trojan Horse Detection", HOST,2008, pp. 8-14.

[38] K. Xiao, X. Zhang, M. Tehranipoor, "A Clock Sweeping Technique for Detecting Hardware Trojans Impacting Circuits Delay," IEEE Design & Test, Vol. 30, No. 2, 2013, pp.26-34.

[39] D. Du, S. Narasimhan, R. S. Chakraborty, and S. Bhunia,"Self-referencing: a Scalable Side-Channel Approach for Hardware Trojan Detection, CHES, 2010, pp. 173-187.

[40] N. Yoshimizu, "Hardware Trojan Detection by Symmetry Breaking in Path Delays", HOST, 2014, pp. 107-111.

[41] Stephan Henzler, "Time-to-Digital Converters", SpringerLink, Volume 29 2010, ISBN: 978-90-481-8627-3 (Print)978-90-481-8628-0 (Online).

[42] Meng-Ting Tsai; Shi-Yu Huang; Kun-Han Tsai; Wu-Tung Cheng, "Monitoring the delay of long interconnects via distributed TDC," in *Test Conference (ITC), 2015 IEEE International* , vol., no., pp.1-9, 6-8 Oct. 2015