On Detecting Delay Anomalies Introduced by Hardware Trojans

D. Ismari, J. Plusquellic
ECE Dept.
University of New Mexico
dismari@unm.edu
jimp@ece.unm.edu

C. Lamech
Intel Corp.
charles.d.lamech@intel.com

S. Bhunia
ECE Dept.
University of Florida
swarup@ece.ufl.edu

F. Saqib
ECE Dept.
Florida Institute of Technology
fsaqib@fit.edu

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. This approach significantly improves the correlation between Trojan-free chips and a simulation-based golden model. Path delay tests are applied to multiple copies of a 90nm custom ASIC chip having two copies of an AES macro. 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. Simple statistical detection methods are used to isolate and detect systematic changes introduced by these additional gates. We present hardware results which demonstrate that our proposed chip-averaging and calibration techniques in combination with a single nominal simulation model can be used to detect small delay anomalies introduced by the inserted gates of hardware Trojans.

Keywords
Hardware Trojan Detection; Path Delay

1. INTRODUCTION

In recent years, semiconductor companies have become fabless, with out-sourced manufacturing. This business model of distributed and out-sourced design, integration, manufacturing, packaging and distribution channels has created challenges related to intellectual property (IP) piracy, netlist and GDSII reverse engineering attacks, integrated circuit (IC) cloning, counterfeit chips and hardware Trojans. 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].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
ICCAD'16, November 07-10, 2016, Austin TX, USA.
@2016 ACM. ISBN 978-1-4503-4466-1/16/11...$15.00.
DOI: http://dx.doi.org/10.1145/2966986.2967061

Unlike 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.

There are alternative techniques for detecting HTs that do not require activation, such as parametric variations analysis. Parametric testing methods provide an advantage over logic-based detection strategies by carrying out a structural analysis of the IC, as opposed to a functional analysis. The advantage of a structural analysis relates to the number of tests that need to be applied to attain sufficient coverage of HTs. Structural testing, focuses on testing each of the elements in the netlist or layout, and therefore, the number of tests is 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. 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 decisions (a decision that it has an HT when it does not).

In this paper, we investigate a parametric detection method that analyzes path delays as a mechanism to detect HTs. An on-chip measurement structure called a time-to-digital converter (TDC) is used to obtain high resolution (approx. 25 ps) 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, and to enhance the correlation between the hardware behavior and simulation models. Experiments are carried out on 44 copies of a custom, 90 nm test chip which incorporate two instances of Advanced Encryption Standard (AES) macro. The contributions of this work are as follows:

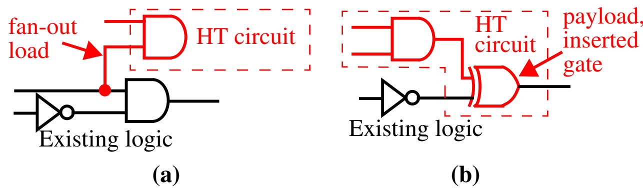

Fig. 1. (a) Fan-out HT gate and (b) payload HT gate.

- The proposed methods are demonstrated using hardware experiments on chips with commercially synthesized implementations of actual functional units.
- An on-chip embedded test structure called a time-to-digital-converter (TDC) is demonstrated as a mechanism of obtaining high resolution path delay measurements.
- A 'tuning' technique for the TDC is described that provides real-time calibration of chip-to-chip variations.
- A **novel chip-averaging technique is proposed and demonstrated in hardware** that nearly eliminates random, within-die variations in path delays, which in turn, significantly simplifies the development of a simulation-based golden model.
- A simple statistical method is used to demonstrate that small delay anomalies can be detected on paths of a HT-modified functional unit, which is confirmed using hardware validation experimental results from a second HT-free copy of the functional unit.
- A simulation-based golden model is derived and used as the basis of the HT detection method.

The remainder 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.

2. BACKGROUND

The insertion of an HT can impact the delay of paths in a circuit in two ways as shown in Fig. 1. A *fan-out* HT gate connects to existing wires in the circuit, and allows the HT to monitor circuit state for activation conditions as shown in Fig. 1(a). The connections add capacitive load to the original path, which in turn impacts delay. The second *payload* HT gate is inserted in series with the original path (see Fig. 1(b)), and is designed to maliciously modify circuit state when the HT activates. The inserted gate typically adds to the path delay in the original circuit and the delay anomaly introduced is typically larger than delay anomalies introduced by capacitive loading effects of fan-out HT gates.

An overview of the approaches that can be taken to detect layout-inserted HTs is presented in [3], and a detailed taxonomy of HTs is presented in [4]. The published work on detecting HTs is too vast to cover, so we focus only on previous work similar to the approach proposed here, which is based on parametric path delay analysis.

The authors of [5] describe a path delay detection methodology which applies principle component analysis to select a reduced dimension path delay data set, a convex hull statistical method for characterizing the HT-free space. In follow-up work in [6], the authors propose a sophisticated golden-model-free statistical method that is based on data collected from a process control monitor. In [7], the authors propose the insertion of shadow registers driven with phase shifted clk for measuring path delays and detecting delay anomalies introduced by HTs. Follow-on work in [8] distinguishes additive Trojan delay characteristics from random variations introduced by process variations.

A technique that configures circuit paths into ROs is proposed in [9] as a means of detecting HTs. An embedded test structure called REBEL, a clock strobing method and simple statistical outlier technique is proposed in [10] as a means of detecting delay anomalies introduced by HTs. The authors of [11] propose the use of test structures as a means of estimating global and within-die process variations, which is used to calibrate path delays. The follow-up work in [12] chooses the shortest paths through each Trojan site as a means of improving resolution. The authors of [13] increase observability of path delays by partitioning the circuit into regions and adding test points. The authors of [14] propose a clock glitching method to measure data paths as a means of authenticating the FPGA IP block and detecting HT anomalies. A clock sweeping path delay measurement strategy is proposed in [15] for detecting HTs.

The authors of [16] propose a golden-free IC method which correlates on-chip sensors with path delays. A golden-model-free HT detection method is proposed in [17] that leverages path symmetries. A pulse propagation technique is proposed in [18] for detecting capacitive loads introduced by HTs on logic paths. The authors of [19] use static timing analysis to detect tampering with the bitstream at runtime on FPGAs. A clock glitching method is proposed in [20] to measure data paths as a means of authenticating the FPGA IP block and detecting HT anomalies. The authors of [21] also propose a clock glitching method to measure path delays and statistical techniques to reduce the adverse effects of inter-die and intra-die process variations. A self-referencing technique is proposed in [22] that compares behaviors of identical functional units, thereby eliminating the need for a golden simulation model.

Given the analog nature of parametric HT detection methods, it is nearly impossible to derive meaningful comparisons among different techniques. The most significant difference between our proposed method and others is the use of *chip-averaging* as a mechanism to deal with the adverse effects of measurement noise and within-die variations on detection sensitivity. A widely recognized challenge of parametric HT detection methods is developing a practical approach for deriving a simulation-based golden-model. Chip-averaging *significantly simplifies* the use of a simulation-derived golden-model by requiring simulation of *only the nominal model* of the design-under-test. In other words, our approach calibrates the hardware data to a single simulation model, as opposed to statistically characterizing the HT-free 'space' using simulation experiments designed to model chip-to-chip and within-die variations.

3. CHIP DESIGN

The test chip layout consists of 2 macros-under-test (MUTs), labeled AES_1 and AES_2 for Advanced Encryption Standard, as shown in Fig. 2. Two copies of an embedded test structure called the time-to-digital converter or TDC (described below) are associated with each MUT [23]. The TDCs are capable of providing accurate delay measurements of signals propagating through the MUTs to its outputs.

A block diagram illustrating the connectivity between an AES MUT, labeled AES_x, and the TDCs labeled TDC_x-1 and TDC_x-2 is shown in Fig. 3. Each TDC has 16 inputs. The first input connects to the clock while the remaining 15 connect to outputs of the

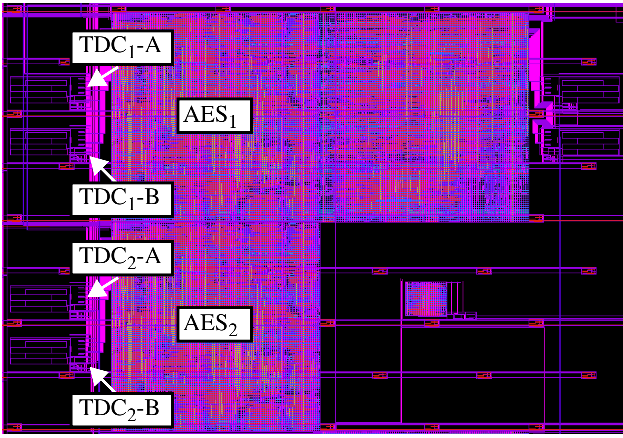

Fig. 2. Chip layout showing AES MUTs and TDCs.

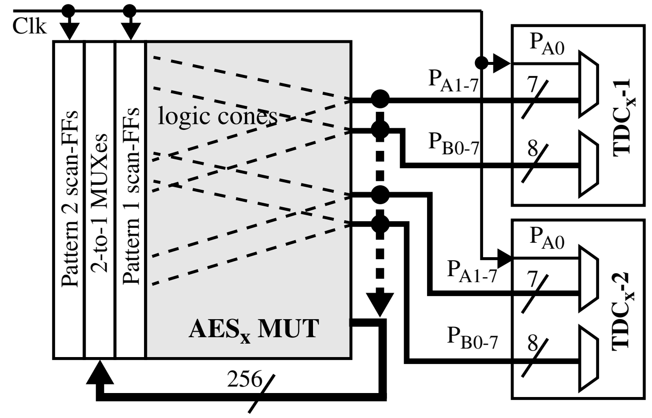

Fig. 3. Block diagram of TDC connections to an AES macro-under-test (MUT).

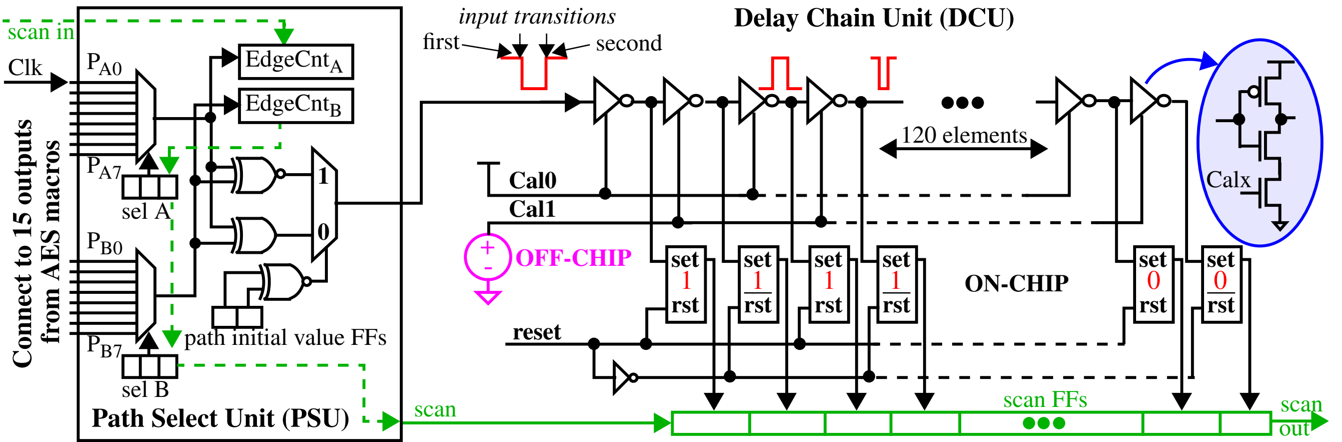

Fig. 4. Pulse Shrinking Time-To-Digital Converter (TDC).

AES MUT. This allows signals propagating through logic cones to 30 of the AES outputs to be timed by the two TDCs[1]. 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 "Pattern 1 scan-FFs" row and the second vector into the "Pattern 2 scan-FFs" row. The Clk can then be used to launch transitions onto the inputs of the MUT.

3.1 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. The TDC is implemented as two components, labeled Path Select Unit (**PSU**) and Delay Chain Unit (**DCU**) in the schematic of Fig. 4. 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 P_{Ax} and one from group labeled P_{Bx}.

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 value FFs" in Fig. 4. These control values select the output of either the XOR or XNOR gate to generate a negative pulse for the DCU (see annotation in Fig. 4). 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 "EdgeCnt$_{A/B}$" 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 reflected 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 '1's followed by a sequence of '0's, represents the

1. The delay chain embedded test structure described in [10] can be used as a practical strategy for providing access to all MUT outputs.

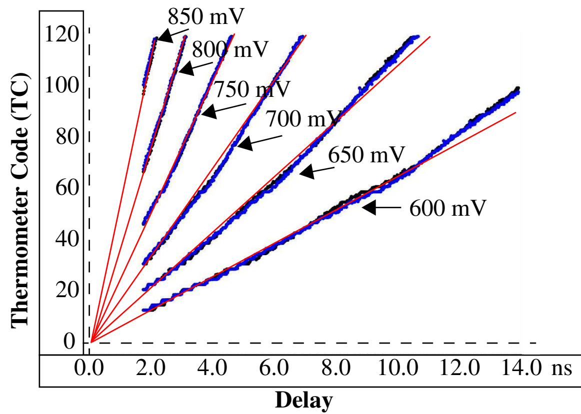

Fig. 5. TDC sensitivity analysis: TC value produced by the TDC (y-axis) as the input pulse width is varied (x-axis) using a variety of Cal1 values (separate curves).

digitized delay difference between the transitions occurring on the paths. A longer sequence of '1's (up to 120 in our version) reflects a larger difference in delay between the two MUT paths.

A call-out of a current-starved inverter used in the delay chain is shown on the far right side of Fig. 4. The NFET transistor with input labeled "Calx" implements the current-starving mechanism. The Calx inputs are driven by two control voltages, labeled "Cal0" and "Cal1". The current-starved inputs of all the even numbered inverters (numbered starting with 0) are connected to Cal0 while the inputs of the odd numbered inverters are connected to Cal1. 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 Cal1 toward the supply voltage allows the odd numbered inverters to switch more quickly. With Cal0 fixed at a specific voltage, larger Cal1 voltages allows the pulse to survive longer in the delay chain because it takes longer for the second edge to catch up to the first 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 Cal0 to the supply voltage. Cal1 is tuned using an off-chip voltage source for each chip in a calibration process described below. The TDC occupies an area of 176 µm x 60 µm (10 kµm^2).

The on-chip integration of the TDC makes it vulnerable to manipulation by an adversary. However, an externally controlled calibration and verification process (both described below) enable any type of tampering by an adversary to be detected.

3.2 TDC Sensitivity Analysis

The Cal1 input allows the timing resolution to be tuned, trading off range and resolution. The curves in Fig. 5 illustrate the trade-off using data collected from one of the 90 nm 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. 4. The DCM allows the width of the pulse to be tuned and stepped with a timing resolution of approx. 18 ps. The x-axis of Fig. 5 plots the pulse width which was varied from approx 1.2 ns (the smallest possible between the FPGA and the test chip board) up through approx. 14 ns. The y-axis plots the TC produced by the TDC for each of the applied pulse widths. The individual curves show the results when using different values for Cal1. For lower Cal1 values, e.g., 600 mV, the timing resolution is approx. 130 ps per TC value while for higher values, e.g., 850 mV, it increases to approx. 20 ps per TC. The calibration process described below tunes Cal1 to values between 750 mV and 850 mV across all chips, which provides approx. 25 ps of timing resolution on average in our experiments.

3.3 Dealing with Process Variations

3.3.1 Calibrating Chip-to-Chip Variations

The ability to tune the TDC using the Cal1 input signal provides a mechanism to virtually eliminate, in real-time, 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. The calibration process is implemented by computing, for a fixed value of Cal1, the average TC from a set of stable paths. A binary search process is incorporated in the calibration process that repeatedly retests these paths, while tuning the Cal1 value, 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 nearly eliminates measurement bias in the on-chip TDCs.

3.3.2 Calibrating Within-Die Variations

Within-die variations currently represent one of the most significant challenges for parametric HT 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 delay of a path from measurements made from all chips (in practice, only a statistically significant sample is required). If within-die variations are random, then the averaging operation will eliminate them, revealing any type of systematic delay anomaly, i.e., an increase or decrease 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 or demonstrated in hardware. Eq. 1 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).

$$\mathrm{CAD}_{A,P} = \frac{1}{\#\,\mathrm{chips}} \sum_{i=1}^{\#\,\mathrm{chips}} D_{A,P} \qquad \textbf{Eq. 1.}$$

3.3.3 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 4 σ limit. The 4 σ 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 actually *unstable* but are able to produce a glitch-free edge and therefore are missed by the EdgeCnt check as discussed in reference to Fig. 4. 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 in the absence of process variations. We also carried out a structural path analysis to discard path delay data for paths that were tested multiple times under different vectors, as a means of removing redundancy in the test data.

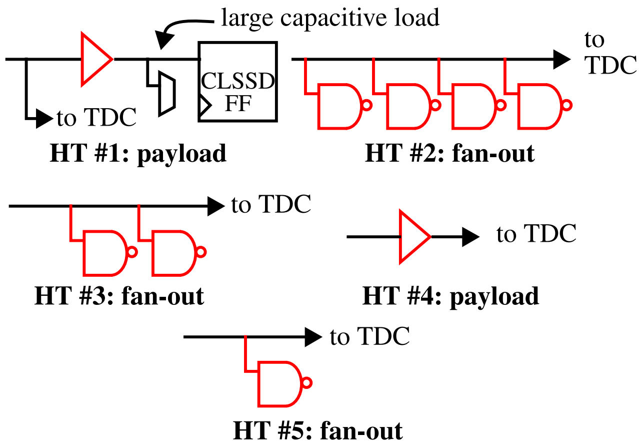

Fig. 6. Layout-editor-inserted fan-out and payload HTs in AES_2.

4. EXPERIMENTAL RESULTS

4.1 Experimental Setup

The objective of our 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. As discussed above, we incorporated two copies of the AES functional unit into the chips, labeled AES_1 and AES_2, as shown in Fig. 2. The two copies are identical except for the insertion of HT gates in the layout of AES_2. AES_1 is included in our analysis as a *validation* vehicle only. The detection method that we propose is based on comparing the path delays measured from Spectra simulation experiments on the *nominal* model of the AES design with the path delays measured from AES_2. Validation is carried out by performing the same analysis but using data from AES_1 instead.

4.1.1 AES Configurations

In order for the validation experiments to be useful, the layouts of AES_1 and AES_2 (including the TDCs) must be identical. This was accomplished by creating one copy of the layout using the Cadence synthesis and place and route tools, and the copying it to two locations as shown in Fig. 2. The layout editor was then used to make small changes to the AES_2 copy. In particular, filler cells were removed and additional gates were added as shown in Fig. 6. The red components represent the HT gates that were added while the black components represent existing gates. **Each HT was designed to effect only one of the AES outputs.** For example, 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.

4.1.2 Hardware Test Sequences

A randomly selected set of 500 test vector pairs are applied to the two AES MUTs simultaneously. All TDC path measurements are made using the Clk as reference. The HTs referred to in Fig. 6 affect TDC_2-2 P_{B3} through P_{B7} inputs while the remaining 11 TDC inputs (8 from TDC_2-1 and 3 from TDC_2-2) are HT-free.

4.1.3 Simulation Golden Model

As discussed above, our technique requires simulations of only the nominal circuit model. The nominal model was created using the Mentor Graphics Calibre XRC extractor. We configured the extractor to produce a resistor-capacitor-transistor (RC) model of the AES_1 (the HT-free macro) and a resistor-capacitor-coupling-capacitor-transistor (RCC) model for the TDC. The coupling capacitors were excluded from the AES_1 extraction because the level of accuracy they added to the simulation results was small, and including them increased simulation runtime significantly. On the other hand, the higher accuracy provided by the coupling capacitors in the simulations of the mixed-signal design-style of the TDC was beneficial. The CADENCE Spectre simulation tool was used to carry out a transient analysis on the AES_1 using a subset of the 500 vector pairs applied in the hardware experiments. Only those vector pairs that provided high path coverage were selected for the simulations.

The data obtained from Spectre simulations of the TDC was used to translate the AES_1 path delays to TCs, using a graph method similar to that shown in Fig. 5 for the hardware. In particular, TDC simulations were performed using a fixed Cal1 voltage and a set of different pulse widths. The line derived by plotting the pulse width against the TC from the TDC experiments was used to translate the AES_1 path delays to TCs.

4.2 Statistical Analysis of the Data

In this section, we develop a simple statistical detection method that targets statistical systematic anomalies in the CAD differences. The systematic anomalies are captured by computing a difference CAD or DCAD value defined by Eq. 2.

$$DCAD_{A,P} = CAD_{1,P} - CAD_{0,P} \qquad \textbf{Eq. 2.}$$

The DCAD analysis can be performed using data from any of the three data sets in pairs, e.g., between the hardware data collected from AES_1 and AES_2, or between each of the hardware data sets and the simulation data. Although the hardware-to-hardware analysis is informative (and is therefore presented below), it is not possible to leverage this type of analysis in practice. The detection technique, instead, is based on comparing the simulation data with the data from AES_2. The analysis of the simulation data against the hardware data from AES_1 is used as validation to confirm the anomalies using the AES_2 data are real and measurable.

For the hardware-to-hardware analysis, 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 $P_{B0\text{-}7}$ TDC inputs as shown in Fig. 3 (Recall that all paths are timed relative to the P_{A0} clk input so only the $P_{B0\text{-}7}$ inputs of the two TDCs can be timed). Each of the groups of DCAD values are plotted as a curve in Figs. 7 and 8. All 16 curves are superimposed in Fig. 7 while Fig. 8 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 five HT TDC inputs, and ensure that all paths driving the remaining 11 HT-free inputs are, in fact, HT-free.

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

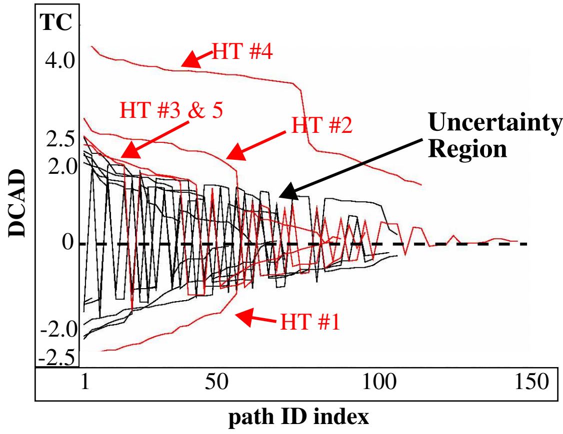

Fig. 7. Hardware-to-hardware DCAD curves for each of the 16 TDC inputs, sorted on the magnitude of the differences from largest (left) to smallest (right).

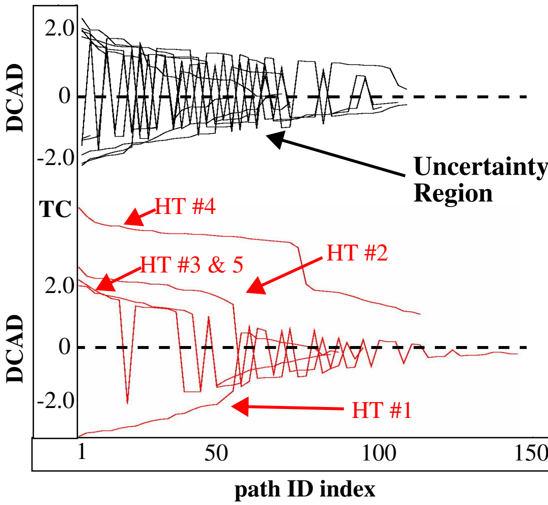

Fig. 8. Fig. 7 with HT-free and HT results separated.

cone-like shape, with a maximum width of +/- 2 TCs on the left, indicates that noise is proportional to the magnitude of the DCAD value, i.e., noise decreases from left-to-right.

Three of the HT curves for HT #1, #2 and #4, are below and above, resp. the Uncertainty Region. They depict a systematic component of variation in path delay, i.e., a shift in delay that is similar in all chips. On the other hand, the curves for HT #3 and #5 are indistinguishable from the noise. The overall behavior of the 5 HT curves is consistent with the expected behavior. For example, HT #1 isolates the large load capacitance of the capture FF and therefore, the delay in AES_2 is smaller than the delay in the AES_1, 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. As noted, HT #3 and #5 are very close to the noise floor suggesting that HTs that create fan-out loads of 1 or 2 gates represent the most challenging HT to detect. However, the curves for the other HTs clearly illustrate the presence of a systematic anomaly.

The results of the proposed technique are shown in Fig. 9, which plots the DCAD values computed using simulation and hardware data from AES_2. Although the differences are always

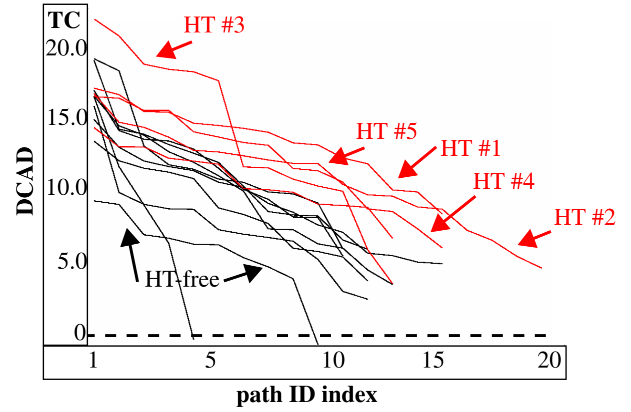

Fig. 9. Proposed Technique: Simulation-to-AES_2 DCAD curves for each of the 16 TDC inputs, sorted on the magnitude of the differences from largest (left) to smallest (right).

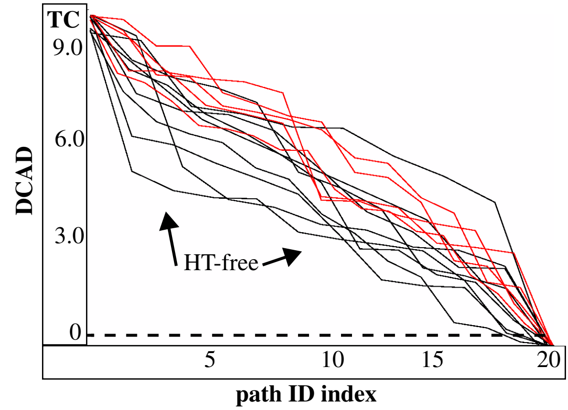

Fig. 10. Validation: Simulation-to-AES_1 DCAD curves for each of the 16 TDC inputs, sorted on the magnitude of the differences from largest (left) to smallest (right).

positive (which is caused by mismatches between the simulation models and the hardware), the anomalous delays introduced by the HTs are evident, e.g., the HT curves are displaced above the HT-free curves. Unlike the hardware-to-hardware analysis, however, the magnitude of the displacement of the curves from the HT-free group is not consistent with expected displacement. This is most noticeable in the curve for HT #4 which should have the largest expected displacement but has instead the smallest. We suspect that mismatch between the simulation model and the hardware is largely responsible for this deviation in expected behavior.

The results for the validation experiment are shown in Fig. 10, which plots the DCAD values computed using simulation and hardware data from AES_1. Since AES_1 does not possess any HTs, there should be no displacement of the HT curves away from the HT-free curves. The HT and HT-free curves are nearly indistinguishable, providing supporting evidence and validation that the displacements of the HT curves in Fig. 9 are related to the delay anomalies introduced by the HTs.

4.2.1 Analysis of the Effectiveness of Chip-Averaging

The effectiveness of chip-averaging in reducing noise and within-die variations is quantified in this section. As indicated above, the CAD values are computed from the TC values measured from 44 chips. Fig. 11 plots the individual chip TCs for a set of 5 HT-free path pairings, labeled P_1 through P_5, as the first 44 values along the x-axis and the CAD value as the last (right-most)

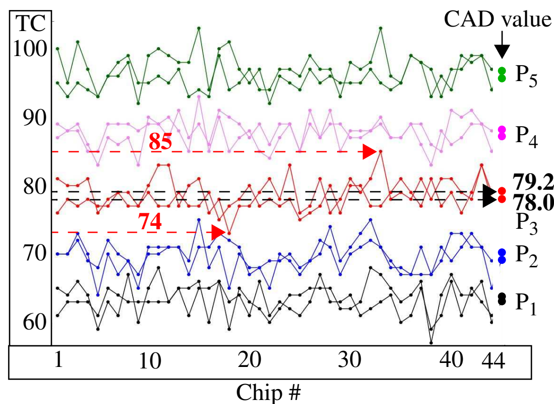

Fig. 11. TCs from individual chips and the chip-averaged delay value (CAD).

data point. The two curves associated with each path pairing are derived from each of the two AES units. The length of the path delay difference increases from P_1 through P_5 as reflected in the range of TCs. The variation in the individual chip values is introduced by noise and within-die process variations. Given all 5 path pairings 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 an order of magnitude, allowing small systematic variations introduced by HTs to be more easily detected.

5. CONCLUSIONS

In this paper, we present chip results derived from a proposed chip-averaging method to detect hardware Trojans (HTs). An embedded time-to-digital converter (TDC) is used to obtain high resolution, approx. 25 ps, measurements of path delays from two nearly identical copies of a hardware instantiation of the AES algorithm. A chip-averaging technique is shown to significantly reduce the adverse effects of within-die process variations on HT detection sensitivity. The technique significantly simplifies simulation-based golden-model development and allows simple statistical methods to be used to detect HTs. The chip-averaging technique is a generalized approach that can be used in any type of parametric HT detection method, as we have shown previously in [24] using I_{DDQ} measurements.

6. ACKNOWLEDGEMENTS

The work is funded in part by National Science Foundation Grants 1603483, 1603475 and 1566530.

7. 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/calendar/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] W. Xiaoxiao, M. Tehranipoor, J. Plusquellic, "Detecting Malicious Inclusions in Secure Hardware: Challenges and Solutions", *HOST*, 2008, pp. 15-19.

[4] R. M. Rad, X. Wang, M. Tehranipoor, J. Plusquellic, "Power Supply Signal Calibration Techniques for Improving Detection Resolution to Hardware Trojans", *ICCAD*, Nov., 2008, pp. 632-639

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

[6] Y. Liu, K. Huang, Y. Makris, "Hardware Trojan Detection through Golden Chip-Free Statistical Side-Channel Fingerprinting", *DAC*, 2014, pp. 1-6.

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

[8] D. Rai and J. Lach, "Performance of Delay-based Trojan Detection Techniques under Parameter Variations", *HOST* 2009, pp. 58-65.

[9] J. Rajendran, V. Jyothi, O. Sinanoglu, R. Karri, "Design and Analysis of Ring Oscillator based Design-for-Trust Technique", *VTS*, 2011, pp. 105-110.

[10] 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.

[11] B. Cha and S. K. Gupta, "Efficient Trojan Detection via Calibration of Process Variations", *ATS*, 2012, pp. 355-361.

[12] B. Cha and S. K. Gupta, "Trojan Detection via Delay Measurements: A New Approach to Select Paths and Vectors to Maximize Effectiveness and Minimize Cost", *DATE*, 2013, pp. 1265-1270.

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

[14] 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 Reconfigurable and Communication-Centric Systems-on-Chip*, 2013, pp. 1-6.

[15] 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.

[16] 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.

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

[18] S. Deyati, B. J. Muldrey, A. Singh, A. Chatterjee, "High Resolution Pulse Propagation Driven Trojan Detection in Digital Logic: Optimization Algorithms and Infrastructure", *ATS*, 2014, pp. 200-205.

[19] G. Sumathi, L. Srivani, M. D. Thirugnana, N. Murali, S.A.V. Satya Murty, T. Jayakumar, "DSDPC: Delay Signatures at Different Process Corners based Hardware Trojan Detection Technique for FPGAs", *Robotics, Automation, Control and Embedded Systems,* 2015, pp. 1-7.

[20] X.-T. Ngo, I. Exurville, S. Bhasin, J.-L. Danger, S. Guilley, Z. Najm, J.-B. Rigaud and B. Robisson, "Hardware Trojan Detection by Delay and Electromagnetic Measurements", *DATE*, 2015, pp. 782-787.

[21] I. Exurville, L. Zussa, J.-B. Rigaud, B. Robisson, "Resilient Hardware Trojans Detection based on Path Delay Measurements", *HOST*, 2015, pp. 151-156.

[22] 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.

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

[24] I. Wilcox, F. Saqib and J. Plusquellic, "GDS-II Trojan Detection using Multiple Supply Pad V_{DD} and GND I_{DDQ}s in ASIC Functional Units", *HOST*, 2015.