# Power Supply Signal Calibration Techniques for Improving Detection Resolution to Hardware Trojans

Reza M. Rad, Xiaoxiao Wang+, Mohammad Tehranipoor+, Jim Plusquellic*
Department of CSEE, Univ. of Maryland, Baltimore Campus
+Department of Electrical and Computer Engineering, Univ. of Connecticut
*Department of Electrical and Computer Engineering, Univ. of New Mexico

## Abstract

*Chip design and fabrication is becoming increasingly vulnerable to malicious activities and alternations with globalization. An adversary can introduce a Trojan designed to disable and/or destroy a system at some future time (Time Bomb) or the Trojan may serve to leak confidential information covertly to the adversary. This paper proposes a taxonomy for Trojan classification and then describes a statistical approach for detecting hardware Trojans that is based on the analysis of an ICs power supply transient signals. A key component to improving the resolution of power analysis techniques to Trojans is calibrating for process and test environment (PE) variations. The main focus of this research is on the evaluation of four signal calibration techniques, each designed to reduce the adverse impact of PE variations on our statistical Trojan detection method.*

## 1 Introduction

A drawback to the globalization of the chip industry is the increased susceptibility of chip design and fabrication to malicious alternations [1][2]. This new threat involves actions taken by an adversary to deliberately modify a chip's design to include a hardware Trojan. The Trojan may be designed to cause the chip to fail at some critical time while operating in mission mode or it may serve to leak confidential information covertly to the adversary. Many types of hardware systems are threatened including those responsible for the security of personal information, and those that implement and support financial infrastructures, military systems and even household appliances. Trojans are cleverly hidden by the adversary to make it extremely difficult for chip validation processes, such as manufacturing test, to accidentally discover them. Many proposed Trojan detection techniques, particularly logic-based testing techniques that target Trojan activation via statistical analysis of unlikely circuit states, will not be effective against even the simplest Trojan hiding techniques.

Most hardware-based security techniques modify hardware to prevent attacks and to protect IPs or secret keys. However, the types of attacks addressed in this paper are fundamentally different. Here the attacker is assumed to maliciously alter the design before or during fabrication. Unfortunately, detection of such alterations is extremely difficult for several reasons. First, physical inspection through destructive reverse engineering is becoming increasingly difficult and costly in nanometer technologies. Second, purposeful activation and discovery of Trojan circuits through application of random or ATPG generated patterns is easily defeated by an adversary. Third, the adversary can configure the Trojan to have a minimal impact on the chip's nominal transient and quiescent leakage current. Therefore, conventional testing methods that measure global, chip-wide, behavior of the power supply current, e.g., $I_{DDQ}$, are ineffective because of very small Trojan-current-to-background-current ratios that are present in multi-million transistor chips.

Given these characteristics, we believe the most effective approach for detecting Trojans is to analyze the parametric properties, e.g., power and delay, across multiple regions of the chip. In particular, we propose a region-based transient power signal analysis method to reduce the impact of increasing levels of process variations and leakage currents. A region is defined as a portion of the layout that receives the majority of its power from a set of surrounding power ports or C4 bumps. In previous work, we showed that regional analysis significantly increases the resolution of power analysis methods to Trojans [3]. However, by itself, it is not sufficient for dealing with the adverse effects of process and test environment (PE) variations on detection resolution. To fully leverage the resolution enhancements available in a region-based approach, such methods must be combined with signal calibration techniques, that are designed to attenuate and remove PE signal variation effects.

In this paper, we propose four signal calibration strategies, one based on quiescent signals (DC) and three based on transient signals (AC), and evaluate them using simulations of a design with inserted Trojans. The improvements to Trojan detection resolution provided by each signal calibration method are determined using a *prediction ellipse* statistical approach and an analysis of outlier residuals. Our simulation results indicate that 1) all signal calibration methods significantly improve Trojan detection resolution of region-based power signal analysis methods when compared with **no** signal calibration, 2) the AC techniques outperform the DC method and 3) there is no significant difference among the three AC alternatives. The last finding is significant because it indicates that a simple sample-based AC calibration method can be used over the more complex waveform sampling techniques.

The remainder of this paper is organized as follows. Related work is described in Section 2 and a Trojan classification scheme is presented in Section 3. The simulated design, inserted Trojans and statistical analysis method are described in 4. Section 5 describes the signal calibration methods and Section 6 presents the simulation results. Section 7 gives our conclusions.

## 2 Background

Security is a major concern in the design and test of chips, particularly in areas that involve protecting secret keys and IPs. The malicious insertion of hardware Trojans in ICs is a new security concern that must now be addressed in combination with conventional security risks. The following summarizes the published work on this topic.

The authors of [4] were the first to address the hardware Trojan issue. They propose the use of side-channel signals, e.g., transient power supply currents, to identify Trojans in chips. The main deficiencies of their technique include the measurement of global signals and the use of signal processing techniques to deal with process variation effects. Also, test environment variations are not modeled or addressed in their work and, unfortunately, are significant detractors for transient signal analysis methods.

The authors of [5] propose an method that first determines a set of target 'hard-to-observe' sites for a Trojan with $q$ inputs and then uses ATPG to generate patterns to activate the Trojan. Although this may be an effective strategy for Trojans with a small number of inputs, analysis complexity and test set size will make this type of approach impractical for larger Trojans.

A Trojan detection method that measures the combinational delay of a large number of register-to-register paths internal to the

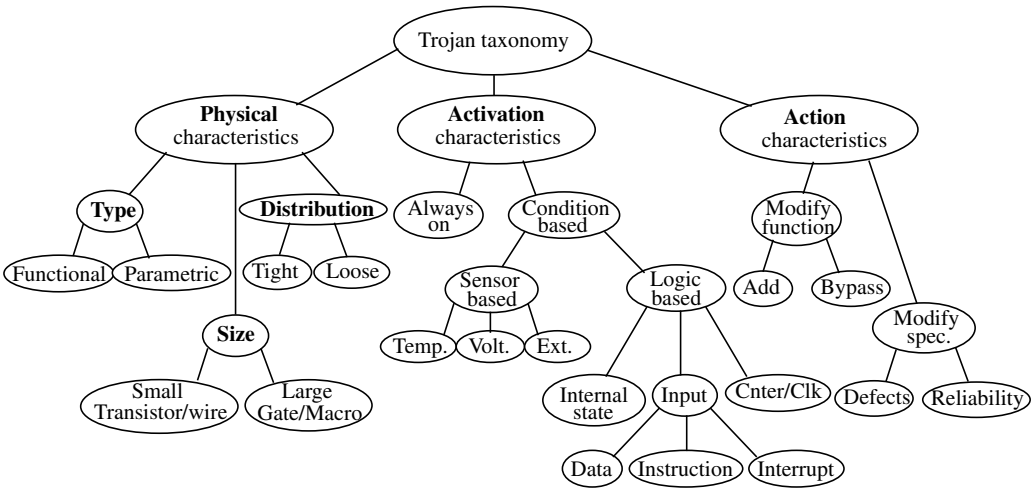


**Figure 1. Taxonomy of Trojans**

functional portion of the IC is proposed in [6]. This approach assumes the attacker is not able to leverage existing 'white space' to insert the Trojan, or use design techniques that avoid adding to path delays. Moreover, the method requires precise characterization of silicon path delays at design time, which is becoming increasingly difficult because of process variation effects.

In [7], the authors propose a region-based stimulation strategy and analyze the global power consumption to detect Trojans. However, their method does not account for PE variations. In [8], the authors introduce special circuitry that enables the direct control of the least controllable nodes in the circuit as a means of triggering the activation of a Trojan. As indicated earlier, methods that target direct activation are practical only for small Trojans. In [9], the authors build a path delay fingerprint of Trojan-free chips by running high coverage input patterns. The number of vectors needed for large designs to achieve adequate Trojan coverage is likely to make this approach impractical.

In previous work, we proposed several region-based $I_{DDQ}$ and $I_{DDT}$ test methods for detecting manufacturing defects and showed that techniques for calibrating PE variations are critical to providing adequate detection resolution [10][11]. The same concern holds true for Trojan detection. In this paper, we compare the previously described DC calibration technique with several new AC calibration techniques and demonstrate the latter are more effective at reducing the adverse effects of PE variations on Trojan detection sensitivity.

## 3 Taxonomy

Our proposed taxonomy describes Trojans using five attributes, including three *physical*, one *activation* and one *action* attribute as shown Figure 1[1]. Trojan physical characteristics refer to the actual layout implementation details of the Trojan. The activation category describes the strategies an adversary may use to trigger the Trojan to carry out its malicious act. The action category describes the types of changes the Trojan may introduce to the IC.

### 3.1 Trojan Physical Characteristics

The *type* sub-category under physical characteristics partitions Trojans into two fundamentally different classes, functional and parametric. The functional class includes Trojans that are physically realized through the addition or deletion of transistors or gates, while parametric refers to Trojans that are realized through modifications of existing wires and logic. The thinning of a wire, the weakening of a transistor or any modification of a physical geometry designed to sabotage reliability or increase the likelihood of a functional or performance failure are examples of the latter. The *size* category accounts for the number of components in the chip that have been added, deleted or compromised while the *distribution* category describes the location of the Trojan in the physical layout of the chip. For example, a *tight distribution* describes a Trojan whose components are topologically close in the layout while a *loose distribution* describes Trojans that are dispersed across the layout of the chip.

### 3.2 Trojan Activation Characteristics

Activation characteristics refer to the criteria that causes the Trojan to become active and carry out its disruptive function. We partition Trojan activation characteristics into two subclasses, labeled *Always-on* and *Condition-based* as shown in Figure 1. Always-on, as the name implies, indicates that the Trojan is always active and can disrupt the function of the chip at any time. This class covers Trojans that are implemented by modifying the geometries of the chip such that certain nodes or paths in the chip have a higher susceptibility to failure. We referred to these types of Trojans as 'parametric' in the *type* subclass of the physical characteristics class.

The Condition-based subclass includes Trojans that are 'inactive' until a specific condition is met. The activation condition can be based on the output of a sensor that monitors temperature, voltage or any type of external environmental condition, e.g., electro-magnetic interference (EMI), humidity, altitude, etc. Or it can be based on an internal logic state, a particular input pattern or an internal counter value. The Trojan in these cases is implemented by adding logic gates and/or flip-flops to the chip, and therefore is represented as a combinational or sequential circuit.

### 3.3 Trojan Action Characteristics

Action characteristics identify the types of disruptive behavior introduced by the Trojan. The classification scheme shown in Figure 1 partitions Trojan actions into two categories; *Modify-function* and *Modify-specification*. As the name implies, the Modify-function class refers to Trojans that change the chip's function through additional logic or by removing or bypassing existing logic. The Modify-specification class refers to Trojans that focus their attack on changing the chip's parametric properties, such as delay. The latter class represents parametric Trojans that modify wire and transistor geometries.

1. Reference [12] elaborates on this taxonomy scheme.

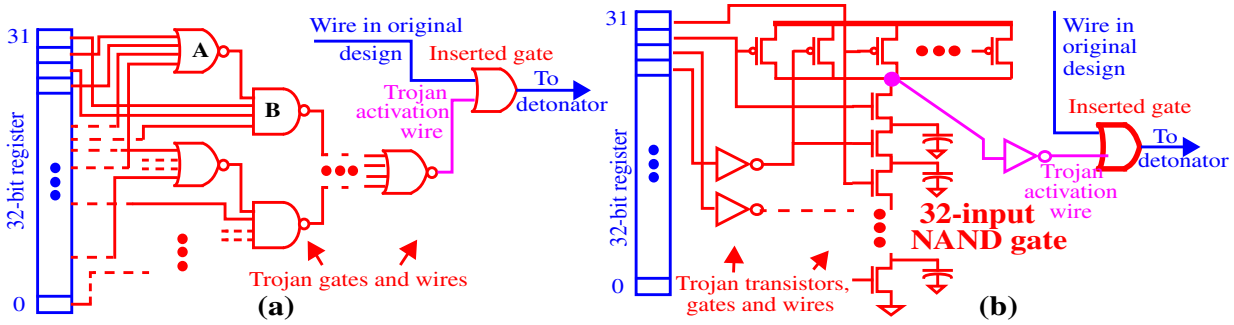


**Figure 2. Multi-stage logic implementation of a Trojan comparator (a) Monolithic logic gate implementation of a Trojan comparator (b)**

## 4 Simulated Design and Trojan Detection Method

Based on the proposed Taxonomy, it is clear that the diversity of Trojans is immense and therefore, the most effective strategy for detecting them may vary depending on the specific characteristics of the Trojan. This paper does not attempt to evaluate the overall effectiveness of region-based transient power signal analysis methods across the proposed Trojan taxonomy (this is left for future work), but rather is focused on determining the most effective method(s) of calibrating transient signals for PE variation effects. The results presented are therefore of benefit to any type of Trojan detection scheme and are independent of Trojan type. However, in order to provide a meaningful context for evaluating the signal calibration methods, a core logic design and several specific Trojan implementations are selected, as described in the following subsections.

### 4.1 Trojan Design

Given the clandestine nature of Trojans, we assume an intelligent and determined adversary will make it nearly impossible to activate the Trojan accidentally or purposefully through functional, structural, and random test patterns during manufacturing test. For functional, condition-based Trojans, the adversary will work diligently to control activation to a time of his or her choosing.

The Trojan implementation shown in Figure 2(a) meets this criteria for a hypothetical military application. The Trojan in this scenario is embedded in a chip that serves as the controller for a missile system. The chip receives encrypted data from a ground-based station through an RF channel and stores the data in a register (shown on the left side of Figure 2(a)). By design, the data is decrypted and checked for validity by core logic components in the chip (not shown).

The gates shown to the right of the register in the figure represent the Trojan. The inputs to the Trojan connect to the register and monitor its state. Since the register holds un-encrypted data, the adversary can control, through his own data transmission tower, the activation of the Trojan at a time of his choosing by transmitting a specific bit pattern to the register. One possible action carried out once the Trojan is activated might be to cause the missile to detonate prior to reaching its target.

The additional circuitry added by the adversary to implement the Trojan necessarily includes some type of comparator to decide when the trigger bit pattern is present. In order to prevent accidental discovery of the Trojan, during, for example, manufacturing test, the comparator must monitor a sufficiently large number of bits and assert its output only on one or a very small number of possible combinations of those bits. For this discussion, assume the values in a 32-bit register serve as input to the Trojan and the comparator asserts on only one set of values. This is sufficient to make it unlikely that the Trojan will be accidentally activated during logic testing, e.g., one chance in $2^{32}$. However, actively monitoring the values in a 32-bit register consumes power, and opens up the possibility that the Trojan may be detected by monitoring the power supply current during testing. Therefore, a secondary issue for the adversary is to minimize the impact of the comparator on the power consumption profile of the chip.

There are many possible ways to implement the comparator. The implementation shown in Figure 2(a) is a multi-stage logic gate with one minterm, i.e., it asserts its output under only one permutation of its inputs. The drawback to this scheme, with regard to power consumption, is the high probability that partial activation of the Trojan will occur during testing. Partial activation is defined when one or more of the NORs and NANDs gate outputs switch in response to changing data patterns in the register. This occurs because each gate monitors only a subset of the register values. The partial activation of the Trojan increases power consumption and makes it possible to detect it. A second scheme is to instead implement the monitor as a monolithic gate, as shown in Figure 2(b). In this case, a 32-bit NAND gate, shown at the transistor level, is used to determine when the target bit pattern is present. Partial activation occurs in this implementation as well by virtue of the inserted inverters and the potential for charging and discharging of the internal parasitic capacitances in the long chain of series n-channel devices.

There are in fact many other implementations possible for the comparator. The main point of this analysis, however, is to demonstrate that Trojans will always have some level of impact on the power profile of the chip. Even if stealthy layout strategies are used that reduce the probability of partial activation, capacitive loading to the wires being monitored will always be present. This is a very important concept because it suggests that it is possible to detect the Trojan without activating it. The challenge is to develop a detection technique that maximizes the sensitivity to potentially small changes in the power profile of a chip with an embedded Trojan. The main detractor to achieving high levels of resolution are process and environmental (PE) variations.

### 4.2 Simulation Models and Detection Algorithm

We propose a power supply transient analysis ($I_{DDT}$) technique for detecting Trojans that is robust to the adverse effects of PE variations. The method analyzes *local*, i.e., within-chip, $I_{DDT}$ measurements obtained from the multiple, individual *power ports* (PPs) on the chip. The method is described following the description of the design and model used in the simulation experiments.

#### 4.2.1 Simulation Model

A block diagram of the IC design used in the simulation experiments is shown in Figure 3. The design includes a six metal layer power grid with nine power ports, labeled $PP_0$ through $PP_9$. The core logic consists of four copies of the ISCAS'85 C499

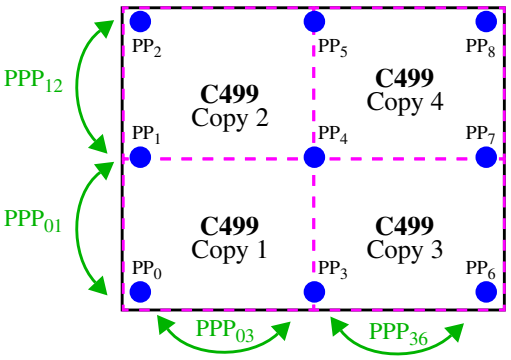


**Figure 3. Architecture of Simulation Model**

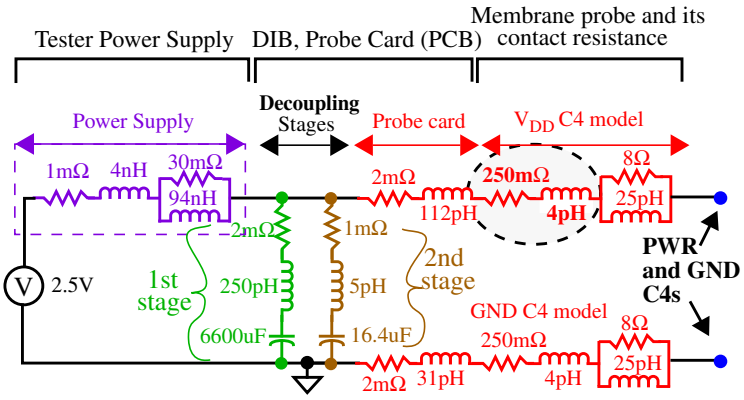


**Figure 4. Probe card model used in the simulation experiments**

benchmark circuit [13]. The design is subsequently referred to as the Quad Core.

The Trojan simulation models are created in two different ways for each of the two Trojans shown in Figure 2(a) and (b). For 2(a), the Trojan is added to the netlist of one copy of the C499 (lower left copy) and CADENCE FIRST ENCOUNTER is used to synthesize the layout. The re-synthesis of the layout with the Trojan gates (shown in Figure 2(a)) introduce significant changes in the placement and routing of the core components of the C499. In contrast, the components of the Trojan shown in Figure 2(b) are added to an empty rectangle that is first introduced in the Trojan-free version of the C499 and the inputs to the Trojan are routed manually to a set of nodes along paths in the C499. Therefore, the differences in the Trojan-inserted and Trojan-free layouts for Trojan 2(b) are small. As shown in the simulation results, these two different implementation strategies make it possible to detect Trojan 2(a) more easily than Trojan 2(b).

PE variations are modeled in two ways. First, twenty unique RC-transistor models of the design are created by configuring DIVA with different sets of TSMC 0.18 um process parameters [14]. The two Trojan-free versions of the layout described above are extracted under each of these process models (PMs) while the two Trojan-inserted versions are extracted under only the first ten PMs. Second, probe card impedance variations are introduced in the off-chip components of the power distribution system. Figure 4 shows a schematic of the power distribution system which includes components representing the power supply (left), decoupling networks and probe card parasitics (middle) and membrane probe card contact parasitics (right) [15]. One copy of the probe card and C4 parasitics are included for each of the nine power ports of the core logic. A dashed circle encloses the resistance and inductance components that are varied in each of the simulation models. The inductor values are varied over 2-20 pH while the resistance values are varied over 0-800 mΩ. The same procedure was used to create twenty Trojan-free simulation models.

Three two-pattern test sequences are used to drive the inputs of the lower left copy of the C499. The inputs to the other copies are held at steady-state. Each of the test sequences sensitizes paths along which some of the Trojan inputs are connected. None of the test sequences cause full activation of either Trojan. However, each sequence introduces different levels of partial activation.

### 4.2.2 Trojan Detection Algorithm

The $I_{DDT}$ detection algorithm is based on a statistical analysis of the $I_{DDT}$ waveform areas generated at the nine power ports as a test sequence is simulated on the Quad Core. For each orthogonal pairing of power ports (see Figure 3), a scatterplot is constructed using the areas produced from simulations of the twenty

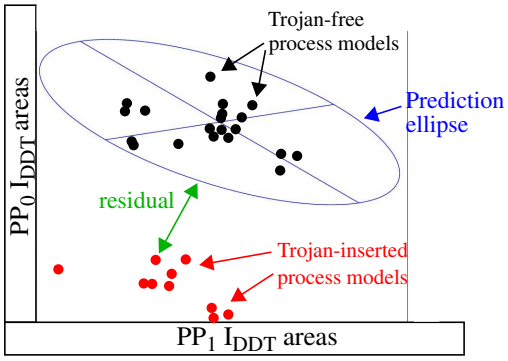


**Figure 5. Example scatterplot for $PPP_{01}$ of Figure 3 for one of the experiments.**

Trojan-free and ten Trojan-inserted PMs. As an example, the scatterplot shown in Figure 5 is created using the areas measured from adjacent power ports $PP_0$ and $PP_1$ for an experiment using Trojan 2(b). A *prediction ellipse* is derived using the first fifteen data points from the twenty Trojan-free PM simulations (the last five Trojan-free PMs are used as control samples.)[1] The dispersion in the Trojan-free data points is a result of un-calibrated PE variations.

Also shown are the data points from the ten Trojan-inserted PMs. For this experiment, the Trojan introduces sufficient regional variation to cause all data point to fall outside the bounds of the prediction ellipse and to be identified as outliers. The residual is labeled for one of the data points in the figure. In our experiments, it is defined as the distance between the data point and the elliptical bound expressed as a standardized quantity. For example, a standardized residual of three indicates the data point is three standard deviations from the elliptical bound.

From the architecture shown in Figure 3, it is possible to construct twelve scatterplots from adjacent power port pairings (PPPs). Several are labeled in the figure as $PPP_{01}$, $PPP_{12}$, $PPP_{03}$ and $PPP_{36}$. For each Trojan-inserted PM, we report the largest standardized residual among the data points that are outliers as a measure of the effectiveness of the signal calibration technique under investigation.

1. The elliptical bound is computed from the eigen values of the covariance matrix and a three $\sigma$ $X^2$ (chi-square) distribution statistic.

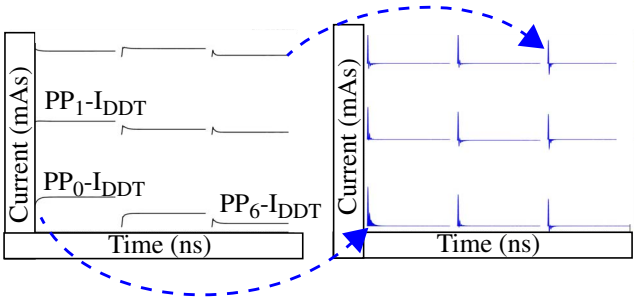


**Fig. 6. Calibration circuit step responses (left) and their integrated derivatives (right).**

## 5 Signal Calibration Techniques

Calibration is used to deal with process variations that occur in the chip's core logic, power grid and off-chip connections to the power ports. It is carried out on each chip using special calibration circuits connected through a scan chain and inserted directly below each of the power ports. The calibration circuits are designed to generate a simple stimulus to the power grid. For $I_{DDT}$ signal analysis, the stimulus is a step current, that is created by enabling a transistor connected between the power grid and ground grid in metal 1.

The left side of Figure 6 gives the simulated step response current waveforms for a calibration test performed using a calibration circuit under $PP_0$ in Figure 3. The waveforms produced at each of the nine power ports are shown at positions that correspond to the layout positions of the power ports (a few are labeled in the figure). As expected, the largest $I_{DDT}$ corresponds to $PP_0$. This is true because the power grid resistance is smallest between the calibration circuit and this power port. Although not shown, eight other calibration tests are performed for each of the remaining power ports, with each test producing a set of nine waveforms as shown for $PP_0$. In total, eighty-one waveforms are produced from the nine calibration tests.

In this paper, we investigate four signal calibration techniques, called DC for $I_{DDQ}$-based, AS for AC-sample-based (first proposed in [17]), AA for AC-area-based and AW for AC-waveform-based, and compare their effectiveness on reducing the adverse effects of PE variations against a fifth technique called NC, i.e., **no** signal calibration. All methods (except NC) make use of the calibration test data to define a transformation matrix. The methods differ in the use of the information available in the calibration response waveforms, as depicted in Figure 7. For DC calibration, the steady-state ($I_{DDQ}$) value of the step response is used. For AS, the value of the step response at 5 ns is used. For AA, the area under the derivative of the step response waveform is used[1]. For AW, the entire step response waveform is used.

Since DC calibration uses $I_{DDQ}$s, it is capable of calibrating for only resistance variations in the CUT and test environment. Therefore, the AC techniques are expected to outperform DC in cases where inductance and capacitance variations are present. This holds true in our experiments because the series inductance in the C4 membrane probe card and the capacitance of the power grids are different in each of the PMs. Among the three AC methods, AS is the simplest because it requires the measurement of only a single sample from the calibration response waveforms. This is similar to the DC technique except that the sample is collected immediately following the introduction of the step input.

1. The derivative of the step response is the impulse response or IR.

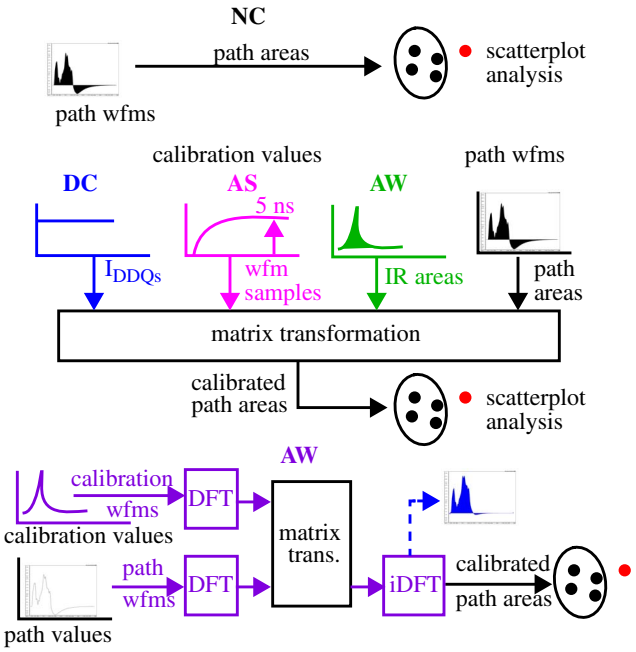


**Fig. 7. Signal Calibration Processes**

The AA technique requires the area under the derivative of the step response or IR waveform to be measured. The measurement instrumentation needed for this technique is more complex and is less attractive for this reason. The AW method requires complete sampling of the step response waveforms and is clearly the most expensive in terms of instrumentation.

The calibration process can be generalized as follows. The data from the calibration tests define a matrix *CM*, as given by Equation 1, where the rows correspond to the calibration tests and the columns correspond to the power ports. Here, $PPD_{xy}$ indicates

$$CM = \begin{bmatrix} \frac{PPD_{00}}{GD_0} & \frac{PPD_{01}}{GD_0} & \cdots & \frac{PPD_{08}}{GD_0} \\ \frac{PPD_{10}}{GD_1} & \frac{PPD_{11}}{GD_1} & \cdots & \frac{PPD_{18}}{GD_1} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{PPD_{80}}{GD_8} & \frac{PPD_{81}}{GD_8} & \cdots & \frac{PPD_{88}}{GD_8} \end{bmatrix} \begin{matrix} \text{cal. test 0} \\ \text{cal. test 1} \\ \cdots \\ \text{cal. test 8} \end{matrix} \quad \textbf{Eq.1.}$$

power port data for calibration test *x* measured at power port *y*. The normalization factor given as the denominator, $GA_x$, is the sum of individual values along each row *x*. A transformation matrix, *X*, is computed from *CM* by taking its inverse, as given by Equation 2, where the elements of *CM*, i.e., $PPD_{00}/GA_0$, are represented as $a_{00}$ in $CM^{-1}$.

$$\underbrace{X}_{} = \underbrace{CM^{-1}}_{}$$

$$\begin{bmatrix} x_{00} & x_{01} & \cdots & x_{08} \\ x_{10} & x_{11} & \cdots & x_{18} \\ \cdots & \cdots & \cdots & \cdots \\ x_{80} & x_{81} & \cdots & x_{88} \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} & \cdots & a_{08} \\ a_{10} & a_{11} & \cdots & a_{18} \\ \cdots & \cdots & \cdots & \cdots \\ a_{81} & a_{82} & \cdots & a_{88} \end{bmatrix}^{-1} \quad \textbf{Eq2.}$$

The transformation matrix is computed for each test chip from the calibration data. Once computed, it is subsequently used to calibrate the path data measured under core logic tests, as given by Equation 3. The vector given by $t_0$ through $t_8$ corresponds to

$$C_n = T_n * X$$

$$\begin{bmatrix} c_0 \\ c_1 \\ \dots \\ c_8 \end{bmatrix} = \begin{bmatrix} t_0 \; t_1 \; \dots \; t_8 \end{bmatrix} \times \begin{bmatrix} x_{00} \; x_{01} \; \dots \; x_{08} \\ x_{10} \; x_{11} \; \dots \; x_{18} \\ \dots \quad \dots \quad \dots \quad \dots \\ x_{80} \; x_{81} \; \dots \; x_{88} \end{bmatrix} \qquad \textbf{Eq3.}$$

the nine data values (areas or waveforms) from the core logic tests. The calibrated data is given by the column vector on the left, i.e., $c_0$ through $c_8$.

The calibrated path data given by $C_n$ in Equation 3 can be used directly in the prediction ellipse method. We refer to this analysis as 'un-normalized' in Section 6. The path data can also be normalized using a process identical to that described for the calibration data, i.e., the elements of the vector $C_n$ are each divided by the sum computed across all elements. We refer to this analysis as 'normalized'.

From Figure 7, the calibration data used in Equation 1 corresponds to single floating point numbers for the DC, AS and AA methods, i.e., current samples for DC and AS and areas for AA. The calibration data for AW is an entire waveform and requires special treatment as shown along the bottom of Figure 7. In this case, a discrete Fourier transform (DFT) is applied to the IR waveforms to convert them into the real and imaginary components appropriate for the matrix inverse operation. The matrix inverse is then computed for each set of frequency components separately. In our experiments, the frequency domain representation contains 1024 real and imaginary components, so 1024 9-by-9 *CM* matrices are constructed and inverted. Normalization is performed by dividing all frequency components by the DC components. The path waveforms are treated in a similar fashion. Once calibrated, an inverse DFT is performed on the calibrated path real and imaginary components and the area under the path waveforms are used in the prediction ellipse method.

One other variant of the calibration process is investigated in this paper. The transformation matrix *X* given by Equation *2* is defined as the matrix inverse of *CM*. It is also possible to calibrate to a specific probe card and process model by multiplying the $CM^{-1}$ obtained for any given process model by the *CM* of the target process model using Equation *2*, i.e., $X = CM^{-1}(\mathrm{PM_a})$ X $\mathrm{CM}(\mathrm{PM_b})$ to calibrate $\mathrm{PM_a}$ to $\mathrm{PM_b}$. The target process model in our experiments is the model identified as *t22t* on [14]. The two approaches are referred to as 'calibrate to IDENTITY' and 'calibrate to t22t'.

The process followed in our experiments is as follows. A set of thirty simulation models are created for the Quad Core, twenty for the Trojan-free design and ten for Trojan-inserted designs. The calibration tests are carried out on each model and the transformation matrix computed. Three core logic test sequences are applied and the $\mathrm{I_{DDT}}$ areas or waveforms are calibrated using the transformation matrix. Twelve scatterplots are created from the calibrated data and the prediction ellipses are computed using fifteen data points from the Trojan-free PMs. The largest residuals for the five remaining Trojan-free PMs and ten Trojan-inserted PMs are computed using the data points that fall outside the elliptical bounds across the twelve scatterplots. The scatterplot analysis is carried out separately using uncalibrated data and using 1) normalized and un-normalized path data and 2) by calibrating the path data to IDENTITY and to t22t, under each of the four calibration methods and three test sequences. This process is repeated for Trojan 2(a) and 2(b).

## 6 Simulation Results

The results of the simulation experiments are displayed in a set of 3-D bar graphs for Trojan 2(a) in Figures 8 and 9 and for Trojan 2(b) in Figure 10. The x-axis of the bar graphs gives the control PMs, C1 through C5, and the ten Trojan-inserted PMs, T1 through T10. The y-axis gives the results for the no signal calibration case in front of the results for the DC, AS, AA and AW calibration cases. The z-axis gives the maximum standardized residuals, i.e., the largest distance among the data points across the twelve scatterplots that fall outside the three sigma limits for each process model. A value of zero indicates that all data points fell within the limits.

The bar graphs of Figure 8 show two results, one for un-normalized data (top) and one for normalized data (bottom). Both bar graphs give results for Trojan 2(a) under the 2nd test sequence with the path data calibrated to t22t. The results in Figure 9 are similar except they are derived from data calibrated to IDENTITY. The results for the 1st and 3rd test sequences in both cases show similar trends and are therefore not shown.

From these results, it is clear that the 'no calibration' technique performs poorly in all cases. For the un-normalized bar graphs, outliers are present in only five of the ten Trojan-inserted process models and all maximum residuals are less than 0.7. In the normalized bar graphs, no outliers are generated so that Trojan is not detected. As described in Section 4.1, the signal variations introduced by Trojan 2(a) are fairly large because of the way it is implemented. This result demonstrates that signal calibration is an important component for achieving a reasonable level of sensitivity to Trojans using transient power supply detection methods.

From the bars corresponding to the calibration techniques, it is clear that each is are able to easily identify the presence of this Trojan. The maximum residuals range from 15 to nearly 60 standard deviations. The failure of the DC calibration method to account for inductance and capacitance variations reduces its sensitivity to Trojans. This is more noticeable in the bottom (normalized) bar graphs where the maximum residuals for DC are noticeable smaller than those for any of the AC techniques. Interestingly, the AC techniques are nearly equivalent in terms of their detection resolution. This is significant because it suggests that the simpler AS technique can be used in place of more complex techniques such as AW which perform full waveform calibration.

The last notable observation concerning these results is the difference in the magnitudes of the maximum residuals of un-normalized and normalized techniques. The normalized technique marginally outperforms the un-normalized technique for this Trojan. There does not appear to be any advantage to calibrating to IDENTITY or t22t.

The bar graph of Figure 10 gives the results across all paths for Trojan 2(b) using the normalized and 'calibrate to t22t' methods. The format is identical to that used in Figures 8 and 9 except for the concatenation of the individual path results along the x-axis. The results for the un-normalized and 'calibrate to IDENTITY' methods show no distinguishable advantage for Trojan 2(b), and consequently are not shown.

The smaller range of values along the z-axis reflects the much smaller signal anomaly introduced by this Trojan, as predicted in the discussion concerning its implementation in Section 4.1. It is also clear that the level of sensitivity to Trojans strongly depends on the test sequence. The maximum residuals of the 2nd test sequence are a factor of nearly three smaller than those under test sequences 1 and 3. Another notable feature is that one of the

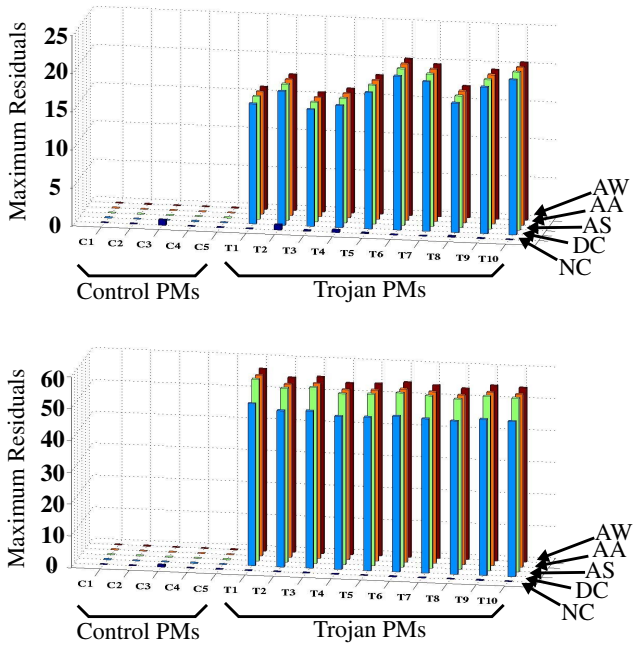


**Figure 8. Maximum residuals for Trojan 1, path 2, calibrated to t22t, un-normalized (top), normalized (bottom).**

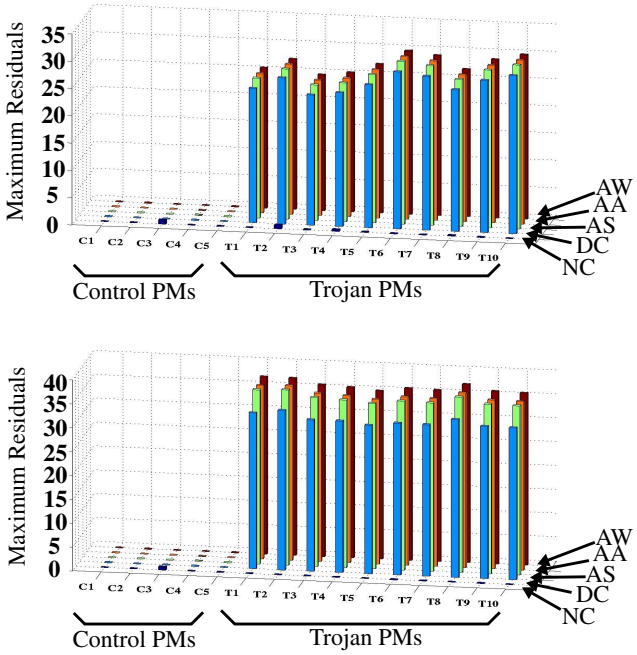


**Figure 9. Maximum residuals for Trojan1, path2, calibrated to IDENTITY, un-normalized (top), normalized (bottom).**

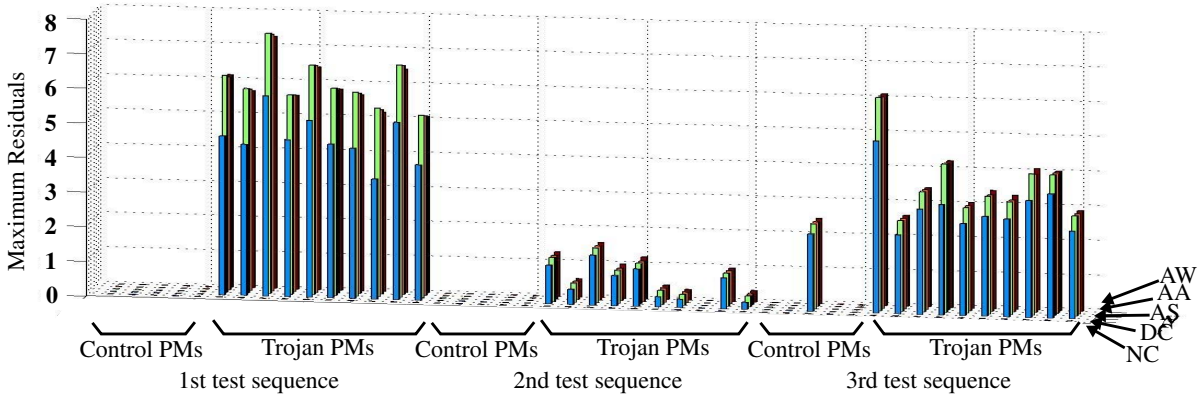


**Figure 10. Maximum residuals for Trojan2, all paths, normalized and calibrated to t22t.**

Trojan-free control PMs produces outliers with a maximum residual of approximately two standard deviations. This indicates the importance of accurately characterizing the Trojan-free process space. The conclusions drawn for Trojan 2(a) concerning the signal calibration techniques hold true here as well. In particular, the method applied without signal calibration for this Trojan produces no outliers under any of the three test sequences.

## 7 Conclusions

A statistical approach is proposed for the detection of Trojans that is based on the analysis of regional transient power supply signals. Four different signal calibration techniques are analyzed to determine their relative effectiveness on reducing the adverse effects of process and test environment variations. Simulation experiments demonstrate that signal calibration is an essential component to enhancing Trojan resolution of the method. The new AC methods proposed in this paper outperform previously described DC methods and the simplest form of AC signal calibration, namely AC sampling, is equivalent in power to more complex schemes. For AC sampling, the important components of the impedance variations in the chip and test environment are captured in a single waveform sample under the condition that the sample is collected close in time (couple of ns) to the delivery of the calibration stimulus.

## 8 Acknowledgements

The work of Reza Rad and Jim Plusquellic was supported in part by NSF grant CNS-0716559. The work of Mohammad Tehranipoor was supported in part by NSF grant CNS-0716535.

## References

[1] http://www.acq.osd.mil/dsb/reports/2005-02-HPMS_Report_Final.pdf

[2] http://www.darpa.mil/mto/solicitations/baa07-24/index.html
[3] R. Rad, J. Plusquellic, M. Tehranipoor, “Sensitivity Analysis to Hardware Trojans using Power Supply Transient Signals”, International Workshop on Hardware-Oriented Security and Trust, 2008, pp. 3-7.
[4] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, B. Sunar, “Trojan Detection using IC Fingerprinting”, Symposium on Security and Privacy, 2007, pp. 296 - 310.
[5] F. Wolff, C. Papachristou, S. Bhunia, and R. Chakraborty, “Towards Trojan-Free Trusted ICs: Problem Analysis and Detection Scheme”, Design, Automation and Test in Europe, 2008, pp. 1362-1365.
[6] Jie Li and John Lach, “At-Speed Delay Characterization for IC Authentication and Trojan Horse Detection”, International Workshop on Hardware-Oriented Security and Trust, 2008, pp. 8-14.
[7] M. Banga and M. S. Hsiao, “A Region Based Approach for the Identification of Hardware Trojans”, International Workshop on Hardware-Oriented Security and Trust, 2008, pp. 40-47.
[8] R. S. Chakraborty, S. Paul and S. Bhunia, “On-Demand Transparency for Improving Hardware Trojan Detectability”, International Workshop on Hardware-Oriented Security and Trust, 2008, pp. 48-50.
[9] Y. Jin and Y. Makris, “Hardware Trojan Detection Using Path Delay Fingerprints”, International Workshop on Hardware-Oriented Security and Trust, 2008, pp. 51-57.
[10] D. Acharyya and J. Plusquellic, “Hardware Results Demonstrating Defect Detection Using Power Supply Signal Measurements”, VLSI Test Symposium, 2005, pp. 433-438.
[11] J. Plusquellic, D. Acharyya, A. Singh, M. Tehranipoor and C. Patel, “Quiescent Signal Analysis: a Multiple Supply Pad IDDQ Method,” IEEE Design and Test of Computers, vol. 23, no. 4, pp. 278-293, 2006.
[12] X. Wang, M. Tehranipoor and J. Plusquellic, “Detecting Malicious Inclusions in Secure Hardware: Challenges and Solutions”, International Workshop on Hardware-Oriented Security and Trust, 2008, pp. 15-19.
[13] http://www.fm.vslib.cz/~kes/asic/iscas/
[14] http://www.mosis.com/Technical/Testdata/tsmc-018-prm.html
[15]D. Acharyya and J. Plusquellic, “Impedance Profile of Commercial Power Grid and Test System”, International Test Conference, 2003, pp. 709-718.
[16] A. Singh, C. Patel and J. Plusquellic, “Fault Simulation Model for iDDT Testing: An Investigation”, VLSI Test Symposium, 2004, pp. 304-310.
[17] M. Sachdev, P. Janssen, V. Zieren, “Defect Detection with Transient Current Testing and its Potential for Deep Submicron CMOS ICs", International Test Conference, 1998, pp. 204-213.