An Experimental Analysis of Power and Delay Signal-to-Noise Requirements for Detecting Trojans and Methods for Achieving the Required Detection Sensitivities

Charles Lamech, Reza M. Rad*, Mohammad Tehranipoor+ and Jim Plusquellic
ECE, University of New Mexico, *CSEE, Univ. of Maryland, Balt.
+ECE, University of Connecticut

ABSTRACT

New validation methods are needed for ensuring integrated circuit (IC) Trust, and in particular for detecting hardware Trojans. In this paper, we investigate the signal-to-noise ratio (SNR) requirements for detecting Trojans by conducting ring oscillator (RO) experiments on a set of V2Pro FPGAs. The ROs enable a high degree of control over the switching activity in the FPGAs while simultaneously permitting subtle delay and transient power supply anomalies to be introduced through simple modifications to the RO logic structure. Power and delay analyses are first carried out across a set of FPGAs using RO configurations that emulate Trojan-free conditions. These experiments are designed to determine the magnitude of process and environmental (PE) variations, and are used to establish statistical limits on the noise floor for the subsequent emulated Trojan experiments. The emulated Trojan experiments introduce anomalies in power and delay in subtle ways as additional loads and series inserted gates. The data from both experiments is used to determine the detection sensitivity of several statistical methods to the transient anomalies introduced by these types of design modifications. A calibration technique is proposed that improves sensitivity to small transient anomalies significantly. Finally, we describe testing techniques that enable high resolution measurements of power and delay to support the proposed calibration and statistics-based detection methods.

1 INTRODUCTION

Hardware Trojans have emerged as a new threat to the security and trust of integrated circuits (ICs). Hardware Trojans are deliberate and malicious modifications to the logic function implemented within digital and mixed signal chips. Hardware Trojans can be designed to shutdown the chip at some pre-determined time and/or when some specific signal or data pattern is received. They may also be designed to remain hidden while leaking confidential information covertly to the adversary. Determining whether a hardware Trojan has been inserted into a chip is extremely difficult for a variety of reasons, e.g., nanometer feature sizes and chip design complexity combine to make optical inspection difficult or impossible.

In this paper, we investigate the impact of specific Trojan implementation characteristics on two parametric parameters, namely power and delay, using FPGAs as a validation platform. The additional logic inserted by an adversary, which represents the Trojan, can be classified into two general forms, the “trigger” and the “payload”. In order to realize the trigger portion, the inputs of the Trojan must connect to existing logic nodes in the IC[1]. These Trojan gate connections increase the capacitive load on these nodes. The payload portion requires change(s) to, or the insertion of a gate in series with, the existing logic of the IC. Both of these modifications change the delay and power characteristics of the IC.

Given the difficulty of designing and fabricating ASICs for this study, we choose to carry out the hardware experiments using the FPGA as a surrogate. Although the implementation of logic paths in a FPGA and a typical ASIC are significantly different, we believe the analysis is meaningful to ASICs because of commonalities in the two platforms, namely, FPGAs, like ASICs, are subject to the same types of process variations, and many of the primitive components of FPGAs, e.g., FFs, lookup-tables, multiplexers, etc. are also found in ASICs. More importantly, our analysis is focused on evaluating the *relative* magnitudes of delay and power variations introduced by noise verses those introduced from Trojan gates, and is therefore less sensitive to the actual circuit implementation characteristics. In fact, the subtle delay variations introduced in the FPGAs to model Trojan trigger connections are relatively smaller than would occur in ASICs. This is true because logic gates implementations in FPGAs have larger overheads in area, power and delay.

Our objective is to evaluate the sensitivity of power and delay analyses to small signal anomalies introduced by the invasive characteristics of Trojan circuit components. The signal anomalies are introduced in a manner that realistically represents the stealthy nature of a Trojan. To simplify the measurement of power and delay and, more importantly, to allow a high degree of control over switching activity, we choose to use a set of ring oscillators (ROs) to represent the core logic of the digital chip under test. The simple structure of an RO also facilitates alternative configurations designed to produce only very small signal anomalies when compared with the default, Trojan-free configuration of the RO. Although this strategy does not emulate an actual Trojan scenario, it is well suited to serve our objective of evaluating sensitivity. More importantly, it decouples our analysis from any specific core logic or Trojan implementation, and therefore, serves as a general evaluation platform to determine the sensitivity requirements of power and delay analysis for Trojan detection.

In the hardware experiments, the FPGAs are configured with 32 copies of a 9-stage ring oscillator (RO), designed as a hard macro and distributed uniformly across the reconfigurable fabric of the FPGAs. A select MUX allows each of the RO outputs to be routed, one at a time, to a high-speed external connector, where an oscilloscope is used to measure its frequency. A high resolution source-meter is used to provide voltage and measure the supply current of, the FPGA’s core logic.

A noise analysis, as well as within-die and chip-to-chip process variation analyses, are carried out on several copies of the FPGA to determine the level of process and environmental (PE) noise. The placement strategy of the 32 ROs is

[1] The exception is fully autonomous Trojan implementations, which we do not consider in this paper.

designed to enable a within-die variation profile to be determined. The analysis shows predominately random variations in RO current and frequency but a strong correlation between the two, suggesting that using the two together may be more effective than using either one alone.

The noise and PE variation analyses are carried out using a specific configuration of the ROs that represents a Trojan-free condition. Once the statistical limits of the noise and PE variation are determined, a variety of invasive Trojan circuit components are investigated, hereafter referred to as 'emulated Trojans'. The emulated Trojans are added to the RO hard macro and are designed to emulate additional capacitive loading and increased delay that a Trojan is likely to add to nodes and paths in the Trojan-free design of an ASIC chip. The additional loading (introduced by the Trojan's trigger connections) is implemented by increasing the fanout of the inverters inside the RO. The fanout gates reduce the RO frequency and increase the power consumption of the RO. The payload portion of a Trojan is modeled as series inserted gates. The 'odd # of inverters' requirement of the RO forces the addition of two inverters, which extends the RO to 11-stages.

Several statistical outlier techniques are used to determine the sensitivity of current and delay analysis for detecting the emulated Trojans. Although it is possible to use each of these parametric parameters to detect Trojans by themselves, we found that using them together is more effective.

A calibration technique is proposed that is able to further enhance the sensitivity of our proposed statistical methods. The proposed calibration technique leverages a set of existing **embedded ring oscillators** (EROs). EROs are inserted by designers into ASIC product chips for the purpose of monitoring process parameters. Given that our experiments are already based on ROs, the calibration technique is easily implemented by selecting a 'calibration RO' from each region (described below). We show that Trojan detection sensitivity can be increased by an order of magnitude over power or delay analysis alone, by using calibration and regression analysis of the measured current and delay parameters.

In the last section of the paper, we describe methods and on-chip support structures designed to facilitate, and improve the sensitivity of, Trojan detection testing methods. In particular, we summarize a previously proposed Multiple Supply Port (MSP) method for obtaining high resolution current measurements and introduce an on-chip Time-to-Digital Converter (TDC) for measuring delay variations with high precision. Unfortunately, neither of these techniques can be implemented, and applied to, FPGAs because of access restrictions and logic circuit limitations of the FPGA platform. Future work will focus on the validation of these methods in an custom ASIC.

2 BACKGROUND

There are several works that investigate process variations in FPGAs. The authors in [1] use at-speed testing to measure the effect of process variations on several 90 nm custom-designed LUT arrays. The authors in [2] measure the effect of both random and systematic process variation on 18 Altera Cyclone II devices with 5- and 7-stage ROs. The authors in [3] measure the level of process variation in 10 FPGAs chip using a 135-stage RO. There is a growing body of work that make use of ROs to define PUFs. For example, the authors of [4] propose two methods to improve the quality of the PUF, a hardware primitive which exploits the manufacturing process variation of the die to create a unique signature, in the presence of the environmental noise and variation in the die. Our work extends this research by investigating noise as well as both intra- and inter-die power and delay variations, and their impact on reducing measurement sensitivity to anomalies introduced by changes in circuit configurations, i.e., emulated Trojans.

FPGAs have been used in several works to investigate Trojans. The authors of [5] design and implement 8 different types of the Trojans on the LUT based FPGA which are unlikely or nearly impossible to detect with functional testing. The authors in [6] propose delay fingerprints defined from a set of test vectors to distinguish Trojans. The authors in [7] propose the use of a shadow latch to detect delay anomalies for the purpose of both authentication and hardware Trojan detection. Our work builds on this research by analyzing noise and process variations in actual hardware, and treats the problem of detecting Trojan anomalies from a statistical perspective. Our proposed power and delay measurement strategies are also regional and self-relative in nature, which distinguishes them from previously proposed methods.

The authors of [8] propose a multi-parameter (dynamic current and maximum frequency) approach for detecting hardware Trojans. Although we also analyze these parameters, the focus of our work is on analyzing within-die and chip-to-chip PE variations and their impact on Trojan detection sensitivity. In [9], the authors decompose the design into blocks and use a self referencing approach for improving sensitivity. We propose calibration techniques that leverage regional correlations of process parameters on the same chip and additionally transforms measured chip data to a single golden model of the design. This strategy significantly reduces the statistical variations introduced by noise and process variations.

3 EXPERIMENT SETUP

The Virtex-II Pro[1] FPGAs under investigation in this work are fabricated in an 130 nm technology. Although process variations in this technology node are not as severe as they are in cutting-edge technology nodes, e.g., 45 nm, our regional analysis approach and calibration methods are designed to scale, and therefore, will remain effective in more advanced technologies.

Figure 1 shows one of the XUP-V2Pro boards, along with a Keithley 2400 and a Tektronix TDS 7254 digitizing oscilloscope. The XUP-V2Pro provides a connector bank and jumpers to allow several components of the board, including the V2Pro chip, to be powered up using alternative external source meters, i.e., the default board power supply can be disconnected. We use a Keithley 2400 high-precision source meter as an alternative power source for the V2Pro chip.

The wiring configuration for the source meter is shown in Figure 2. The Keithley is setup in 4-wire (sense) mode which allows the power terminals close to the board to be maintained at 1.5 V, the core supply voltage for the V2Pro. This mode effectively eliminates the IR drop across the power wiring from the Keithley to the board.

A MUX is configured into the reconfigurable fabric of the FPGA to enable one of the RO outputs to be routed to a pin on the V2Pro, and then off the XUP board through a

[1] Mounted on Digilent development boards

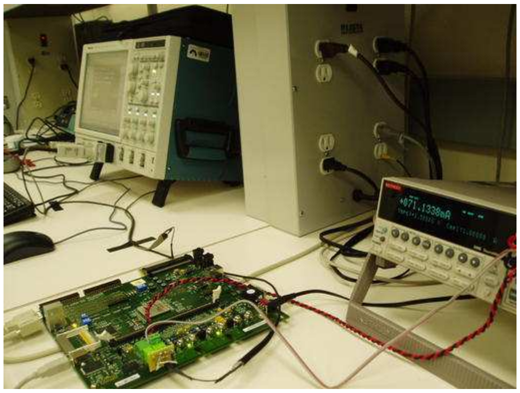


Fig. 1. XUP V2-Pro board and experimental setup with Keithley 2400 source meter and Tektronix TDS 7254 digitizing oscilloscope.

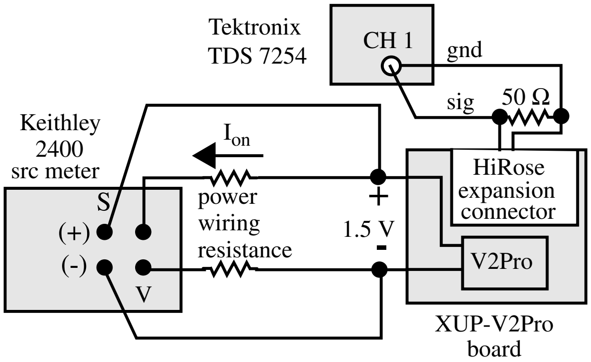


Fig. 2. Custom connections to the XUP-V2Pro board for current and frequency measurements.

high-speed HiRose expansion connector. A 50 Ohm termination resistor is connected between signal and ground on this connector pin as shown in Figure 2, and an active probe with an input bandwidth of 1.5 GHz (Tek P6245) is used to measure the frequency of the pulse train generated by the RO on the oscilloscope.

4 EXPERIMENTAL DESIGN

A 9-stage RO is used as the primary design component in the hardware experiments, and is shown in Figure 3. The RO is configured with an 'enable' pin to allow any one or more of the 32 ROs on a given chip to be enabled. An 'output enable' pin is also included to prevent the RO pulse train from driving the output routing support circuitry. This is required to obtain accurate current measurements for the ROs because the output routing support circuitry impacts the power and on-chip temperature significantly when enabled.

Figure 4 shows the support circuit components configured into the FPGA. We use a serial port (UART) to set the select inputs of the *RO-enable MUX*. The 32 outputs of the RO-enable MUX are connected to the *enable* inputs of the NAND gates of the 32 ROs (see Figure 3). The select MUX is designed such that more than one RO can be enabled simultaneously. The *output-select MUX* routes exactly one of the RO outputs to the output pin as shown on the HiRose connector in Figure 2. LABVIEW software was used to completely automate the data collection process.

Figure 5(a) shows the placement of the 32 ROs in a block level diagram of the V2Pro. The ROs are placed on 4 diagonals from the corners of the chip to the center. The RO in the upper left corner is designed as the **reference RO** (to be discussed). The support circuits, e.g., UART and MUXes, are shown along the bottom of the figure. Figure 5(b) shows a callout of the RO hard-macro, which includes a set of 5 CLBs and associated routing.

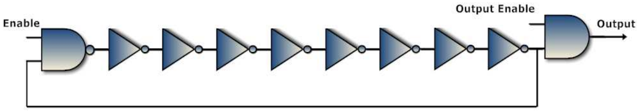


Fig. 3. Schematic of the 9-stage RO.

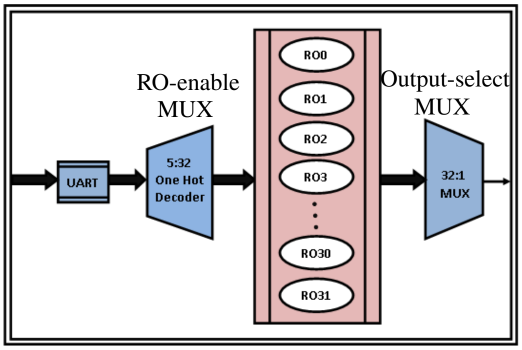


Fig. 4. FPGA design components added as support for controlling the ROs.

4.1 Emulated Trojans

The Trojan scenarios that we investigate fall into two broad categories: 1) Trojans that add capacitive load and therefore increase delay and power consumption, called '**trigger Trojans**' and 2) those that add series inserted gates, which increase delay but have a smaller impact on power, called '**payload Trojans**'. Each of these models the nature of actual Trojans in ASIC chips, where the trigger portion of the Trojan is connected to existing circuit nodes for the purpose of activation and the payout portion is inserted into logic paths for the purpose of modifying logic functions.

Figure 6 shows the modifications made to the base RO shown in Figure 3 to emulate the two Trojan types. The trigger Trojan adds additional capacitive load by leveraging the **intrinsic fanout** that already exists in the CLB blocks. Figure 7 shows a screen snapshot using Xilinx's FEdit of a CLB with labels on two of the input wires. For the Trojan-free tests, the upper input is used in the implementation of the RO, and for the trigger Trojan tests, the lower input is used. Larger capacitance loads are implemented by repeating this implementation for other CLBs in the RO, e.g., four CLBs are modified this way to implement Trojan #4 shown in Figure 6. The payload Trojan (Trojan #5) shown in Figure 6 is implemented by adding two additional CLBs (inverters) to the base RO implementation.

Our initial attempt to emulate a trigger Trojan added capacitance load by connecting additional CLBs to nodes in the RO using the switch fabric of the FPGA. Although this strategy increased the level of measured current, frequency was largely unaffected. Closer examination of the FPGA revealed that the switch resistance effectively isolates the additional capacitive load from the driving inverter in the RO, and is not a good model for fanout in an ASIC.

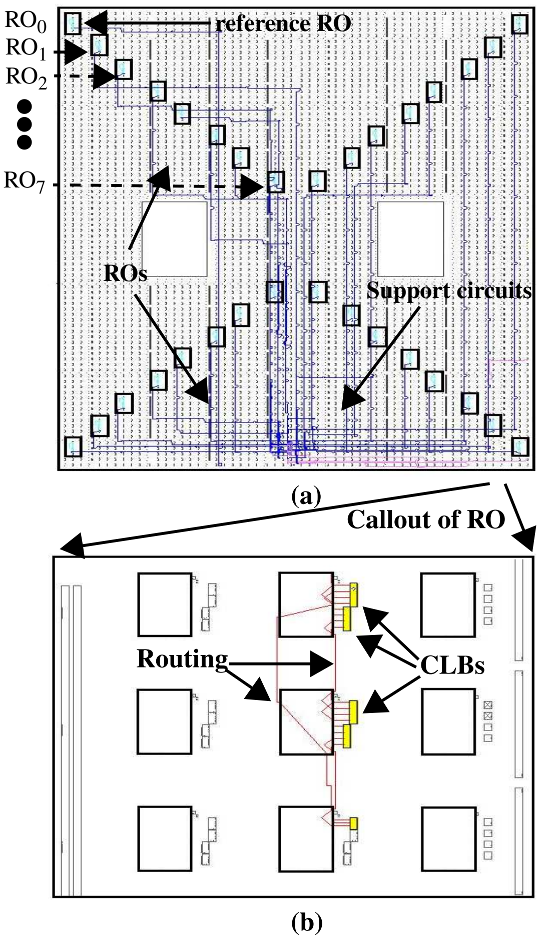


Fig. 5. (a) FPGA architecture with embedded hard-macro Ring Oscillators (ROs) and support circuits, (b) Ring Oscillator (RO) hard macro design, showing the 5 CLBs and connections that implement the 9 stages.

Our choice to leverage the natural fanout already present in the wiring configuration of the CLB, on the other hand, is a close match to an actual loading condition introduced by a Trojan in an ASIC, and therefore, is a better Trojan emulation strategy.

Each of these Trojans is implemented in a separate RO and is used to replace the Trojan-free version of RO_0 (the reference RO) in 5 different FPGA designs. Therefore, a total of 6 configurations are implemented, one Trojan-free (TF) and 5 Trojan (T1, T2, etc.) implementations.

5 EXPERIMENTAL RESULTS

The experiments were carried out on 20 copies of the XUP-V2Pro board. The first set of experiments are designed to determine the impact of noise and process variations on RO power and delay as a means of establishing statistical limits for the subsequent emulated Trojan experiments. Both noise and process variations (within-die and chip-to-chip) work to reduce the sensitivity of delay- and power-based Trojan detection methods, and are increasing in magnitude in nanometer technologies. Therefore, it is critical to measure and evaluate them. The results of this analysis demonstrate that methods designed to calibrate for these sources of variations are essential in achieving meaningful sensitivity.

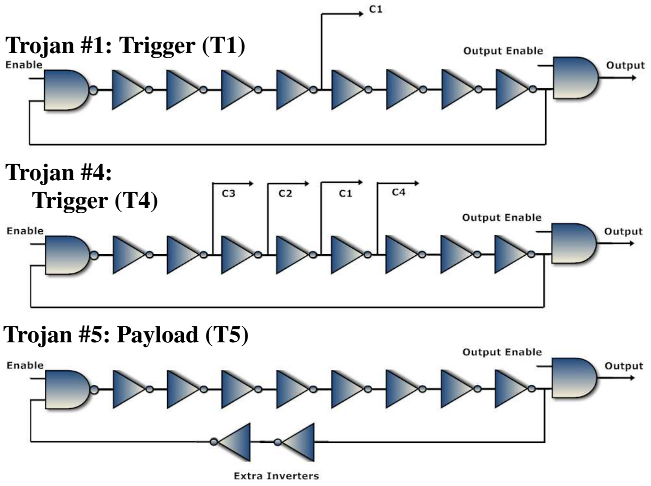


Fig. 6. Modified RO schematics showing implementations of two 'trigger' and the 'payload' components of the emulated Trojans.

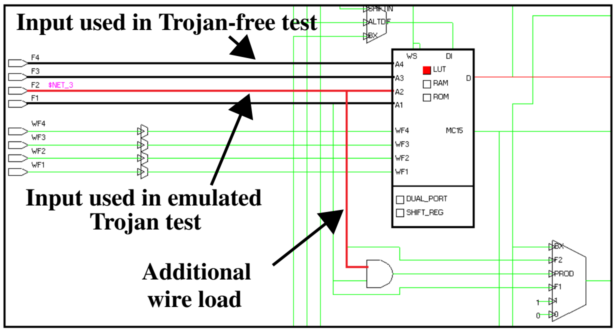


Fig. 7. Xilinx FEdit view of CLB showing 'trigger Trojan' implementation.

5.1 FPGA Analysis

5.1.1 Noise and Within-Die Variation Analysis

In order to get a handle on the level of uncertainty in the experimental evaluation process, we carried out a 'noise' analysis on several copies of our chips. For each chip, a sequence of 32 *null* current and RO current measurements were made using the Keithley. In order to isolate the current in a RO from that of the support circuitry, the *output enable* in Figure 3 was set to '0'. Therefore, the support circuitry remains idle (not switching) during all of the current measurements. None of the ROs were enabled for the null current measurements. Immediately following each null current measurement, a RO was enabled and the current was again measured. For each RO, the **net current** is computed by subtracting the *null* current value from the value measured with the RO enabled. This process effectively eliminates the chip-wide DC leakage currents and minimizes temperature variation effects.

We also measured the frequencies of the ROs in a separate process that followed the global current measurement phase. Separating global current measurements from frequency measurements minimized the adverse effects of on-chip temperature fluctuations introduced by the output support circuitry. These temperature fluctuations added significant levels of noise to the global current measurements. For the frequency measurements, the *output enable* was set to

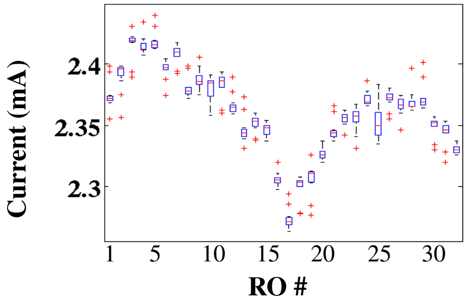


Fig. 8. Noise and intra-chip variation: Boxplot of current behavior from Chip B1 for 32 ROs.

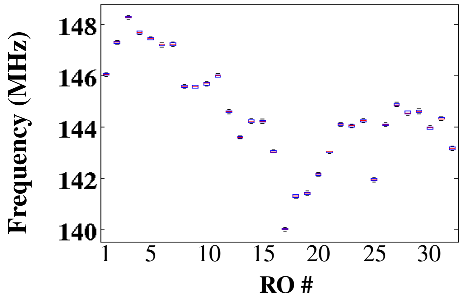


Fig. 9. Noise and intra-chip variation: Boxplot of frequency behavior from Chip B_1 for 32 ROs.

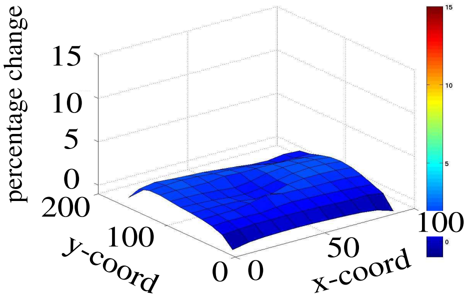


Fig. 10. Chip-to-Chip variation: Mean current percentage change (PC_{mx}) w.r.t. reference RO_0.

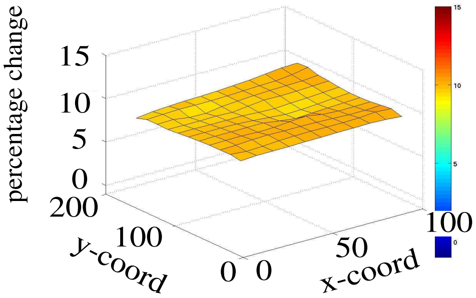


Fig. 11. Chip-to-Chip variation: three σ percentage change in current (PC_{sx}) w.r.t. reference RO_0.

'1' and the oscilloscope was set to average 128 samples of the output waveform. This entire sequence of experiments was repeated 12 times on several boards for the noise analysis.

The boxplots in Figures 8 and 9 summarize the statistics computed using the 12 data sets of current and frequency measurements, respectively, for chip B_1 for each of the 32 ROs. The ROs are listed along the x-axis. The noise statistics are summarized by 6 values on the y-axis in each box plot: the medium, the upper and lower fence limits (for largest and smallest observations, respectively), upper and lower quartiles, and outliers. The first important observation is that the magnitude of noise is smaller than the within-die variation, particularly for frequency. For example, the set of standard deviations associated with the boxplots reflect the **measurement noise** for each RO, and their average can be used as a measure of noise across the entire analysis. The average standard deviation for current is approx. 9 uA and for frequency, it is 50 KHz. Similarly, **within-die variation** is reflected in the standard deviation of the means across the 32 boxplots in each figure. For current, the standard deviation is 34 uA and for frequency, it is 2 MHz.

A second important observation in the data is the correlation between current and frequency, which can be seen by comparing corresponding ROs in each figure. The Pearson correlation coefficient measures the level of correlation between two data sets. The correlation coefficient computed using the 32 mean currents from Figure 8 against the 32 mean frequencies from Figure 9 is 94.4%, where 100% represents perfect correlation.

Average case analysis of the noise works well to show trends. However, worst case analysis is needed to properly determine the sensitivity of power and frequency for detecting Trojans. The worst case 3 σ values of the noise are 53 uA and 332 KHz, and as **percentage change**[1], are approx. 2% and 0.2%, respectively. Likewise, the worst case 3 σ values for within-die variation are approx. 4-5% for both current and frequency. These worst case values have significant bearing on defining the statistical limits of the noise, which we will show in the Trojan analysis section below.

5.1.2 Chip-to-Chip Variation Analysis

The behavior of the mean values shown in the boxplots of Figures 8 and 9 for one chip is different for each of the other chips. In this section, we analyze the chip-to-chip variation in the 20 FPGAs and show that it is predominately random with a large variance. Unlike the noise analysis, only one sample of current and frequency from each chip is used in the chip-to-chip analysis. The chip-to-chip variations in current are presented and those for frequency are summarized.

A simple strategy for analyzing chip-to-chip variations is to construct a 20x32 matrix of the RO currents (or frequencies) from the chips, with the rows representing chips and the columns representing ROs. From this matrix, the mean values of the currents in the 32 columns are then computed. Since the mean is computed across the chips for each RO, any systematic trend that is common to all chips will produce a pattern across the set of mean values. On the other hand, if within-die variation is random across chips, the mean values will be very similar (no pattern).

To show this and to enable comparisons between the

[1] Percentage change is defined below and computed w.r.t. reference values of 2.25 mA and 140 MHz as given by the minimum values Figures 8 and 9.

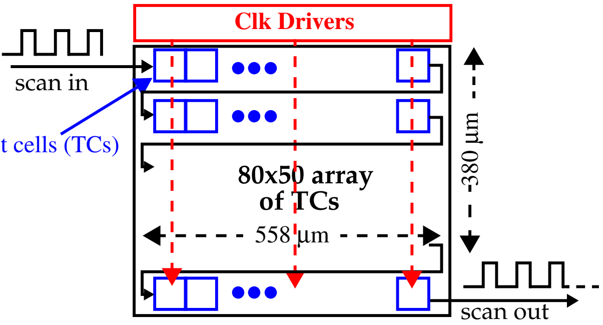


Fig. 12. Block diagram of test structure's scan path.

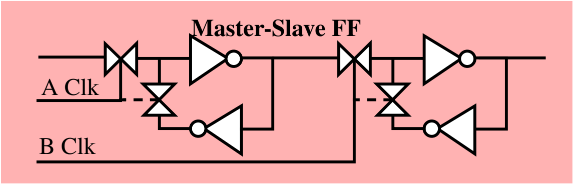


Fig. 13. Scan FF with A/B Clks to enable flush delay.

current and frequency analyses, the mean values of the columns of the matrix are first converted to percentage change (PC_{mx}) values. PC measures the relative difference in current with respect to a *reference value*. The mean value computed for RO_0 (first column) is used as the reference in this analysis. Eq. 1 gives the expression for PC using mean currents. Here, I_x represents the mean current for RO_x, and I_0

$$PC_{mx} = \frac{(I_x - I_0)}{I_0} \times 100 \qquad \textbf{Eq. 1.}$$

is the mean current from RO_0.

Figure 10 plots the PC_{mx} values on the z-axis of a 3-D surface plot. The x-y plane represents the x-y plane of the chip, with the coordinates given in units of CLBs. The distribution of the 32 ROs, as shown in Figure 5, provide a 'sampling' of the values in the x-y plane, with the remaining values linearly interpolated from these measured values[1]. The fact that the entire surface is close to 0% indicates that chip-to-chip variations are primarily random. A small systematic trend is visible as a 'dimple' at (x,y) = (50,100), but the magnitude of PC_{mx} values remains small over the entire (x,y) region of the chip with bounds of -1.0% to 2.5%.

A similar procedure is used to measure the second component of chip-to-chip variation, i.e., the standard deviation. Using the matrix described above, the standard deviations are computed across the columns, as we did for the means. Eq. 2 is used to compute a PC_{sx} for each of the standard deviations σ_x. σ_x is multiplied by 3 to scale the standard deviation so that it includes 99.73% of the population.

$$PC_{sx} = \frac{3\sigma_x}{I_0} \times 100 \qquad \textbf{Eq. 2.}$$

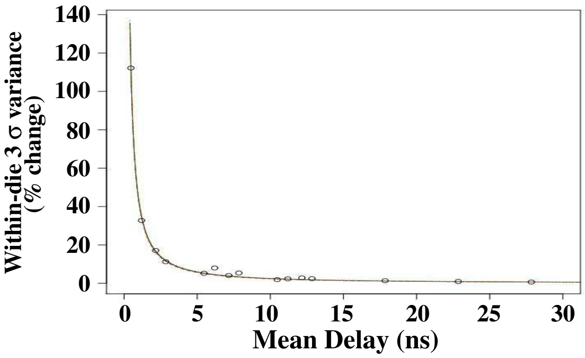


Fig. 14. Within-die delay variations expressed as % change against mean path delay in ns on the x-axis for a set of 65 nm test chips.

The PC_{sx} values for the 20 FPGAs are plotted in Figure 11, using the same format as that described for Figure 10. The entire surface 'hovers' around 10%, again with small deviations between 9% and 11%. The featureless characteristic of the surface supports the random nature of chip-to-chip variations and illustrates that within-die current variations are fairly uniform in magnitude across the chips at approx. 10%. The frequency analysis yielded similar results with bounds of -1% to 2% for PC_{mx} and 11% to 12.5% for PC_{sx}.

5.2 Analysis of Delay Variations in ASICs

In order to validate the need for calibration methods further, we analyze delay variations in a set of eighteen 65 nm test chips to illustrate the trend of increasing within-die variations. A block diagram of the test chip is shown in Figure 12. The test-chip consists of an 80x50 array of test circuits (TCs) connected together through a scan chain. Each of the 4,000 TCs contains three master-slave FFs for a total of 12,000 FFs. The master-slave FFs are designed in an LSSD fashion, with separate clocks driving the master and slave latches as shown in Figure 13. The dual clock configuration allows a long delay chain to be created by setting both clocks high. Since each FF has two pass-gates and two inverters in series, the delay chain is effectively 48,000 gates long (12,000 FFs x 4 gates/FF).

A sequence of experiments were carried out in which a rising edge is launched into the scan chain input at time t_0 = 0 ns and at a time t_x, the A clk is de-asserted to stop the propagating edge. With the scan chain initialized to all 0s, the number of 1's captured in the scan chain indicates how far the edge propagated over the t_x time interval. We conducted a sequence of experiments on each of the chips in which the launch/capture delay t_x was varied from 500 ps to approx. 1200 ns in 5 ns intervals, i.e., approx. 240 experiments were carried out per chip.

The analysis shows that the chip-to-chip variation in delay along the entire chain is approx. 15% (data not shown). However, **within-die** variations are much larger, particularly along shorter segments of the delay chain, as shown in Figure 14. Here, the mean delays along various segments of the delay chain are computed using the 18 chips and are plotted along the x-axis[2]. The 3 σ variations in the delays of these path segments are plotted along the y-axis as percentage change. For example, the within-die variations for the 5 ns segment are approx. 10%, i.e., delays

[1] We use the surface plots and linear interpolation only as a visual aid, and do not intend with these plots to predict the behavior in regions that are not sampled.

in the range of 4.5 to 5.5 were measured across the 18 chips. The trend in the data is captured by the superimposed exponential curve. For path delays of 500 ps (approx. 6 master-slave FFs as shown in Figure 13), the % change increases to approx. 100%[1]. Although the chip-to-chip variation is only slightly larger than the 11-12% observed in the 130 nm FPGA technology, this analysis shows that the regional, within-die variation is significantly larger and needs to be accounted for in Trojan detection methods.

5.3 Trojan Analysis

A second important objective of this work is to determine the impact that Trojans have on power and delay, and to investigate analysis methods that are sensitive to small anomalies in these parameters. As discussed earlier, an FPGA platform is used to model an ASIC because of the flexibility it provides in reconfiguring the logic to model different Trojan scenarios. We recognize that the current and frequency characteristics of equivalent ROs in an ASIC will be different, i.e., smaller power consumption and faster in frequency (as demonstrated in the previous section for delay), but so would the power and frequency anomalies that are introduced by the modeled Trojans. Therefore, we believe our experimental approach and analysis, although an approximation, will scale to ASICs.

The within-die and chip-to-chip variation analysis carried out in the previous section is essential for determining the types of analysis methods that may be effective in detecting Trojan anomalies. Methods based on power and delay analysis are inherently statistical in nature. The bounds of the statistical limits that separate Trojan-free and Trojan-implanted chips must be derived from the level of measurement noise and process variations present in the chips.

5.3.1 Statistical Analysis Methods

In this section, we describe several statistical methods designed to detect the emulated Trojans. A simple 1-dimensional (1-D) statistical method is used to analyze RO current and frequency data in isolation. A 2-D method, called regression analysis, is used to analyze current and frequency together, to determine the advantage, if any, of combining both parameters for detecting Trojans. Last, a calibration method is proposed that is designed to further improve the detection sensitivity of the statistical methods.

5.3.2 1-D Analysis

A 1-D statistical analysis is carried out using the currents and frequencies from RO_0. Recall that RO_0 is used to emulate each of the 5 Trojans in separate hard-macro designs of the RO, as shown in Figure 6. A set of 3 σ statistical limits are computed from the current and frequency data measured with each of the 20 FPGAs configured with the Trojan-free version of RO_0. The plots in Figure 15 show the Trojan-free data on the left side as a series of 'vertically stacked' data points, called a line graph, for current (a) and frequency (b). The line graph of the Trojan-free data is labeled with 'NT' on the x-axis. The y-axis plots current and frequency, respectively. The 20 data points in the NT line graph are used to define three lines labeled *mean*, *upper 3 σ limit* and *lower 3 σ limit*. The factor of 3 is used to scale the standard deviation (σ) such that 99.73% of the sample population is included between the limits. The value of 3 σ is commonly used in industry as a statistical limit.

The line graphs labeled 'T1', 'T2', etc. depict the data from the Trojan experiments. Several trends are notable. The currents are larger in the (a) plot for all Trojans, in comparison to the NT currents. Also, there is an incremental increase in current for the trigger Trojans T2, T3 and T4 over T1, as expected given the increase in capacitive load added to the RO by these Trojans. Payload Trojan T5, implemented as 2 additional series-inserted inverters in the RO, also increases the current over the NT case. The frequency plot (b) shows the opposite trend, i.e., the trigger and payload Trojans increase delays and reduce the frequencies of the ROs. It is also clear that the delay added by the payload Trojan is more significant than that added by the trigger Trojans.

Given the large dispersion in the Trojan-free data points, only some of the Trojans are actually detected. We consider a Trojan detected if its data point falls above or below the 3σ limit lines. The number of detections for each Trojan are tabulated on the right side of the plots. For example, none of the trigger Trojans, T1 through T4, are detected in the frequency analysis.

5.3.3 Calibration

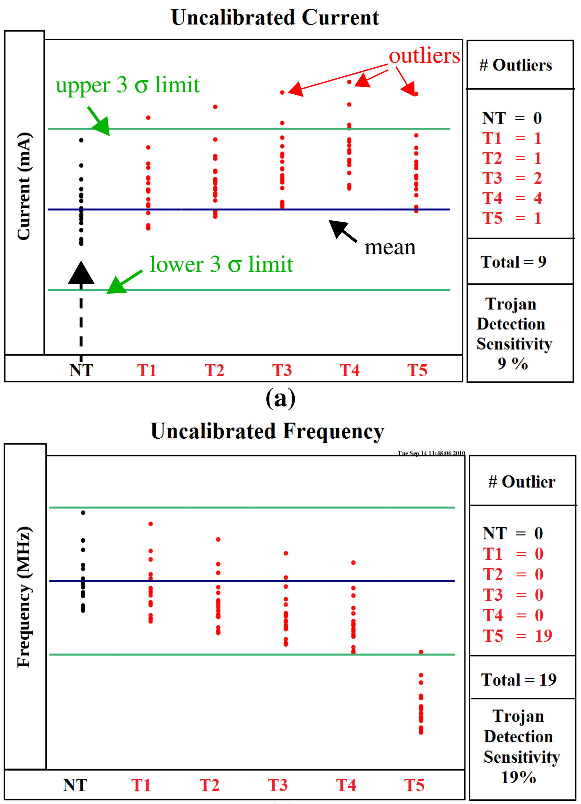
The large dispersion in the Trojan-free (NT) data points in Figure 15 is caused by measurement noise and chip-to-chip process variation effects. The overall effect of these noise sources is to reduce the sensitivity of statistical methods to Trojan current and frequency anomalies. Calibration can be used to reduce chip-to-chip (and within-die) process variation effects.

We propose a 'regional' calibration technique that uses the current and frequency measured from a 'calibration RO' to help compensate for process variation effects. This type of calibration assumes that a distributed set of ROs are embedded into the product chip. As discussed earlier, embedded ROs are increasingly inserted into ASIC designs for monitoring process variations and wear-out effects, and therefore, our proposed scheme simply leverages these existing resources. The basic idea is to compute a current or frequency ratio using, in our case, the reference RO_0 and a 'calibration RO' that is located in the same region. In an ASIC, the measured current or path delay from an applied logic test would be used instead of the values measured from the reference RO. In either case, the ratio is used to calibrate for within-die and die-to-die variations in current and frequency.

In our experiments, we partition the FPGA layout as shown in Figure 5 into 4 regions, each of which contains 8 of the 32 ROs. Our regional calibration method uses one of the ROs in each region for calibration. RO_3 is chosen as the calibration RO for calibrating current and frequencies for circuit activity that occurs in the upper left region of the chip, which is the region in which the emulated Trojans are inserted. The calibration process involves computing the ratio of the current (or frequency) generated by RO_3 on a

[2] Chip-to-chip variations are eliminated from this analysis by scaling the delays of each chip by a factor, computed as the ratio of the delay from a large t_x experiment (t_x is approx. 1000 ns) on each chip and the delay from a reference chip.

[1] A full analysis of these experimental results will be presented in a future work.



Outliers

NT = 0
T1 = 1
T2 = 1
T3 = 2
T4 = 4
T5 = 1

Total = 9
Trojan Detection Sensitivity 9 %

(a)

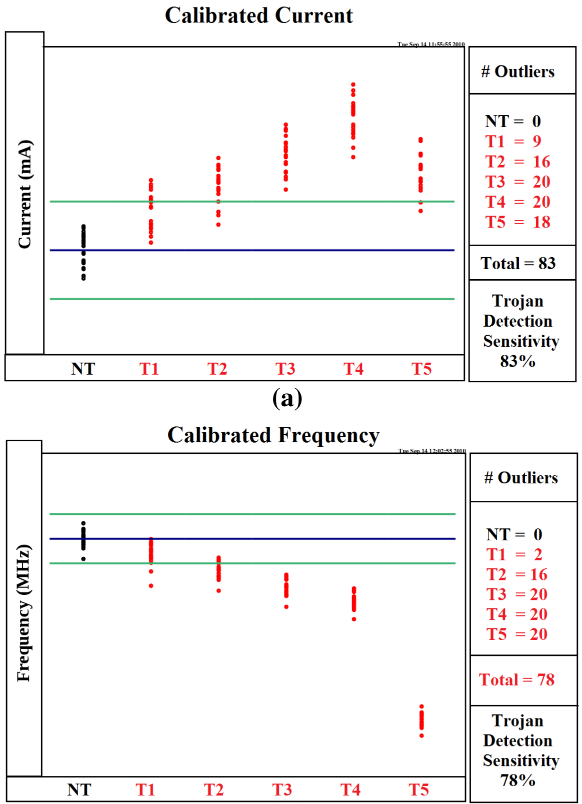
Outlier

NT = 0
T1 = 0
T2 = 0
T3 = 0
T4 = 0
T5 = 19

Total = 19
Trojan Detection Sensitivity 19%

(b)

Fig. 15. (a) Uncalibrated current and (b) frequency 1-D statistical analysis using 20 Trojan-Free RO_0 values (NT), and 20 RO values from each of the 5 Trojans (T_x).



Outliers

NT = 0
T1 = 9
T2 = 16
T3 = 20
T4 = 20
T5 = 18

Total = 83
Trojan Detection Sensitivity 83%

(a)

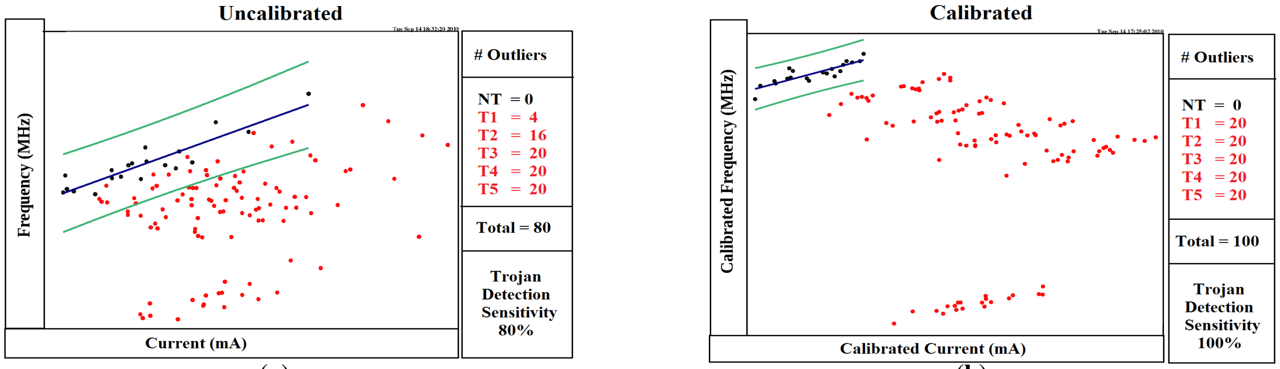
Outliers

NT = 0
T1 = 2
T2 = 16
T3 = 20
T4 = 20
T5 = 20

Total = 78
Trojan Detection Sensitivity 78%

(b)

Fig. 16. (a) Calibrated current and (b) frequency 1-D statistical analysis using 20 Trojan-Free RO_0 values (NT), and 20 RO values from each of the 5 Trojans (Tx).



Outliers

NT = 0
T1 = 4
T2 = 16
T3 = 20
T4 = 20
T5 = 20

Total = 80
Trojan Detection Sensitivity 80%

(a)

Outliers

NT = 0
T1 = 20
T2 = 20
T3 = 20
T4 = 20
T5 = 20

Total = 100
Trojan Detection Sensitivity 100%

(b)

Fig. 17. (a) Uncalibrated and (b) calibrated regression analysis using 20 Trojan-Free RO_0 values (NT), and 20 RO values for each of the 5 Trojans (Tx).

reference chip (B_1 is used in our experiments), to the values measured from RO_3 on each of the remaining chips. These ratios are used to multiply the currents (or frequencies) measured from other 'regional' RO experiments on these chips, in particular, those from RO_0.

Figure 16 shows the data from Figure 15 after the calibration process is performed. It is clear that the dispersion in the Trojan-free data (and the corresponding limits) is significantly reduced, i.e., the 3 σ limit lines are closer together. The number of detections also increases significantly over the uncalibrated data analysis, as given by the tabulated results on the right side of the plots. For example, total detections increase to 83 and 78 for current and frequency analysis, respectively, over the uncalibrated results given in Figure 15 as 9 and 19.

5.3.4 Regression Analysis

A second statistical analysis method that we investigate is regression. Regression analysis is carried out on a scatterplot that contains data from two chip parameters. For our experiments, we plot frequency against current, as shown in Figure 17, by pairing the current and frequency values from RO_0 across the 20 chips. Since current and frequency are correlated, the data points from the Trojan-free experiments cluster around a line, called the regression

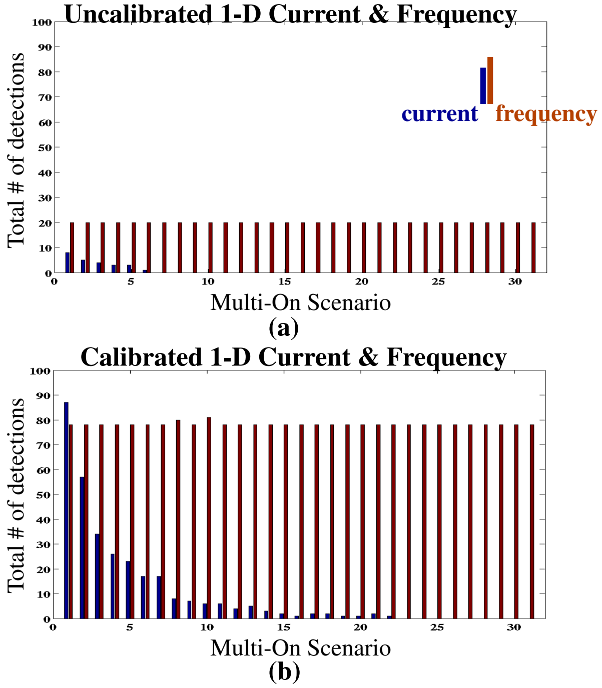


Fig. 18. Total Trojan detections using uncalibrated (top) and calibrated (bottom) current and frequency data for Multi-On experiments.

line. A regression line is derived from the Trojan-free data points and represents the best fit line through them. A set of statistical limits can also be defined in an analogous fashion to the line plot analysis described earlier. The regression line and 3 σ parabolic limits using uncalibrated and calibrated data are shown in the (a) and (b) plots of Figure 17, respectively. Trojan detection is also defined in an analogous manner as data points that fall above or below the limits.

Observations similar to those given earlier can be made regarding the dispersion of the data points for both plots. The main point to this analysis is to determine if Trojan detection sensitivity is better when both current and frequency information is used together (as opposed to being used alone as in the 1-D analysis). This is clearly true in the uncalibrated data analysis where the number of total detections increases from 19 (the larger of the two values in Figure 15, i.e., from the frequency data) to 80, as given in the (a) plot of Figure 17. Using calibrated data, the detection sensitivity increases further, from 83 as given in the (a) plot of Figure 16 to 100, i.e., all Trojans detected, in the (b) plot of Figure 17.

5.3.5 Multi-On RO Experiments

The results from the current analysis of the RO_0 experiments described in previous sections are best case in the sense that only one RO is enabled. In an actual ASIC application, it is extremely difficult or impossible to control logic activity such that only a single path is sensitized (propagates a edge). It is far more common to apply a test that causes multiple paths to propagate edges. Therefore, the single RO experiments are not representative of an actual testing scenario. In this section, we investigate Trojan sensitivity using 1-D and regression analysis when more than one RO is enabled. We call these scenarios Multi-On.

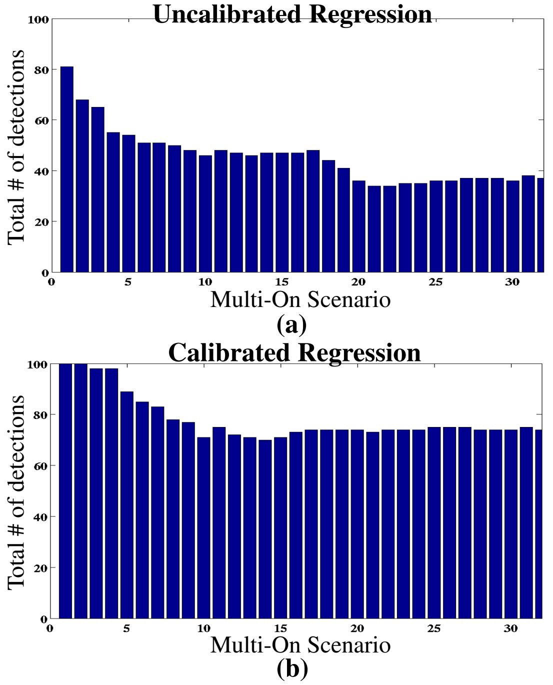


Fig. 19. Total Trojan detections using uncalibrated (top) and calibrated (bottom) data under regression analysis for Multi-On experiments.

The Multi-On experiments are numbered from 1 to 32 to indicate the number of simultaneously enabled ROs. RO_0, which contains the Trojan in the Trojan experiments, is always enabled. Multi-On scenario 2 enables RO_0 and RO_1 (see Figure 5 for RO positions). Multi-On scenario 3 enables RO_0 through RO_2, etc. As more ROs are enabled, the level of 'background' current increases. We define background current as the current generated from the additional ROs, i.e., RO_1, RO_2, etc.

Figures 18 gives the total number of detections in a bargraph using uncalibrated (a) and calibrated (b) data for the 1-D analysis method. The Multi-On scenario is given along the x-axis. The results for current and frequency are shown as adjacent bars under each scenario. The number of detections goes to 0 using uncalibrated current for Multi-On scenarios larger than 6, which illustrates that current sensitivity drops dramatically as the background current increases[1]. The same behavior is observable under the calibrated current analysis (bottom of Figure 18), except that Trojans continue to be detected up through Multi-On scenario 22. Intuitively, the Multi-On scenarios have a much smaller impact on the frequency analysis because the frequency anomalies introduced by the Trojan remain distinct and relatively independent of the amount of additional switching activity.

The regression results are shown in Figure 19. Only one bar is present for each Multi-On scenario because regression uses both current and frequency data as described earlier. The total number of detections are larger than zero across all Multi-On scenarios using the uncalibrated data (plot (a)) because frequency remains effective

[1] We discuss a regional current analysis method in Section 6 that is able to preserve sensitivity.

under the Multi-On scenarios. A similar trend is observable in the (b) plot in Figure 19 which shows the regression results using calibrated data. The average height of the bars is larger, however, confirming again that calibration has a significant impact on detection sensitivity. Interestingly, the regression analysis using calibrated data (Figure 19(b)) is slightly worse than the calibrated frequency data (Figure 18(b)) for several of the Multi-On scenarios in the range of 10-15. The difference is small but illustrates that the additive effect of noise from two data sets, as is true for the regression analysis, can reduce sensitivity.

6 DISCUSSION

It is clear from the results presented in Section 5 that a high resolution analysis of current and delay, combined with a calibration method to attenuate the adverse effects of within-die and die-to-die process variation effects, has the potential of providing high sensitivity to the small current and delay anomalies introduced by Trojans. Our experiments were designed to determine the advantages of such a strategy under highly controlled conditions. Unfortunately, in practice, it is not possible to obtain high resolution measurements of current and delay without some type of support infrastructure on-chip and/or in the manufacturing test environment. In this section, we describe test strategies and an on-chip support infrastructure that can be used to accomplish these goals.

In previous work, we proposed a multiple supply port method (MSP) as a means of obtaining high resolution transient current measurements [10]. Subsequently, we also demonstrated in hardware experiments on a 65 nm test chip a similar strategy using leakage current (I_{DDQ}) [11]. However, as shown in this work, transient current analysis becomes increasing less effective as the number of simultaneously exercised paths increases. This is particularly evident for global current measurement methods, but MSP will be adversely impacted as well. Therefore, maintaining the effectiveness of transient current methods requires a special form of automatic test pattern generation (ATPG) that is designed to target specific paths and minimize transient 'background' noise, i.e., the total number of simultaneously exercised paths. We describe a method for achieving this in a recent work [12].

Also as demonstrated in this work for some of the Multi-On scenarios, the highest level of Trojan detection sensitivity is achieved by combining current and delay analysis. Unfortunately, conventional delay test methods used in manufacturing test cannot provide high resolution measurements of delay of individual core logic paths. This is true because the launch/capture timing used in delay tests is fixed throughout the test application process and therefore, only an 'upper' bound on the delay of **all** tested paths is determined. Although modifications of the test application process have been proposed where the launch/capture timing is changed dynamically and repeatedly for each test pattern until tight upper bounds are determined for **each** of the tested paths, such methods are difficult implement in practice because of cost constraints imposed in manufacturing test, the limitations of automatic test equipment (ATE), masking of invalid bits, etc. We believe the same problems will exist for the process of detecting Trojans using delay tests.

Given these limitations, a better solution to obtaining high resolution delay information is to incorporate an on-chip infrastructure to support it. The authors in [13] describe one such infrastructure, which makes use of shadow registers to capture subtle delay variations in paths introduced by Trojans, and analyze its effectiveness using simulations on a variety of process models. We describe here a novel infrastructure designed to accomplish this goal, and which attempts to do so in a more cost effective manner, using an infrastructure with a smaller footprint and compatibility with existing ATE capabilities.

The alternative infrastructure that we propose for measuring delay anomalies introduced by Trojans is shown on the left in Figure 20. Designs of this type are called *flash time-to-digital converters* (TDC) and are used in applications such as light detection and ranging (LIDAR) [14]. The TDC is designed to provide *relative* delay information of two logic paths in the chip-under-test by creating a digital code that represents the time difference (Δt) between the two edges. These values can be used in place of the frequencies in the calibration and statistical techniques described in Section 5.

The inputs to the TDC are given as $\mathbf{I_1}$ and $\mathbf{I_2}$, on the top left in the figure. These inputs are MUXed onto a pair of path end points, e.g., the inputs of the Capture-FFs of a pipeline register, in the chip-under-test. The edges that arrive at these end points drive the inputs of the TDC's XNOR and XOR gates. The displacement of these edges in time creates a negative pulse on the output of either the XNOR or XOR gate whose width is proportional to the Δt between the edges. The *neg pulse select* signal is set to select the output which produces the negative pulse, i.e., is set to 0 if both edges transition in the same direction (as shown) or to 1 if they transition in opposite directions[1].

This pulse is then converted to a digital value as follows. First, the chain of FFs is initialized using scan to all 0's. The *measure/scan* input is then set to 0 and a test is applied to the chip-under-test. The difference in the timing of the edges along the core logic paths generates a pulse that propagates along the inverter chain. As the pulse moves to the output node of each inverter, it is also routed to the clock input of the corresponding FF, which flips the FF state from 0->1. This action is repeated as the pulse moves down the inverter chain until the pulse disappears (see right side of figure).

The rate at which the pulse shrinks is controlled by the *calibration* analog control pin. By appropriate sizing of the NMOS and PMOS transistors of the inverter chain, the *calibration* input can be 'tuned' to slow down the front edge of the pulse such that the back edge eventually catches up to it. Once the test has been applied, the values of the FFs in the chain define a thermometer code, i.e., all 1's in left-most FFs followed by all 0's in right-most FFs. The digital value that corresponds to the width of the pulse is given by the position in the chain in which the FF values change from 1->0. The scan chain is used to clock out the thermometer code after each applied test to the core logic.

The scan chain is also used to scan out the FF bits for two types of calibration. For *range* calibration, the thermometer code is used to 'tune' the analog voltage controlling the pull-up path resistance of the inverters as a means

[1] This architecture cannot distinguish which edge arrived first. However, simple modifications to the TDC can be made to handle this, which are beyond the scope of this paper.

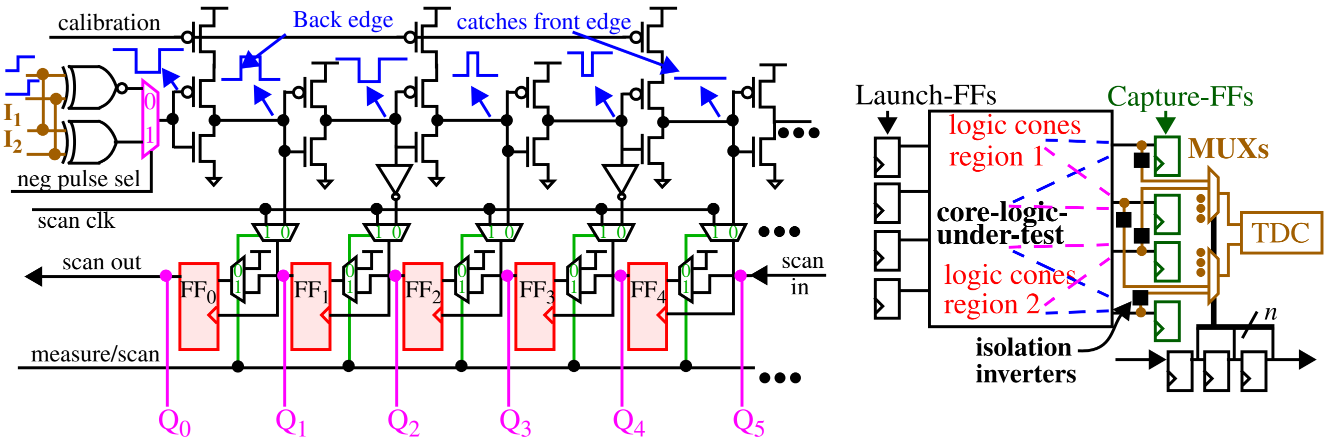


Fig. 20. Time-to-Digital Converter (TDC) for measuring regional delay variations introduced by Trojans (left), integration of TDC on-chip (right).

of adjusting the range of measurable Δt's. For *timing* calibration, the objective is to learn the relationship between a fixed Δt and the value of the thermometer code (for a given analog voltage) for each chip-under-test. To accomplish this, external instrumentation, e.g., ATE, applies a fixed Δt to a pair of I/Os that can be MUXed onto the I_1 and I_2 inputs (see below). In addition to allowing a thermometer code to be converted to an actual Δt, another important driver for learning this relationship for each chip is to allow process variations that occur within each TDC to be accounted for. Converting the thermometer codes to actual Δt's is essential for achieving a high level of sensitivity to Trojan delay anomalies.

We propose to connect the TDC to the core-logic-under-test using a MUX scheme as shown on the right in Figure 20. This allows the relative delays of paths terminating at pairs of Capture-FFs to be measured while minimizing the overhead associated with the TDC scheme, e.g., only one TDC is required per macro. The scan chain along the bottom of the figure is used to select paths to compare. The MUXing scheme is designed to allow paths from different regions (logic cones) to be compared, labeled as *region 1* and *region 2* in the figure. This strategy is designed to improve the chances that only one path in the pair is affected by a Trojan, which, if true, increases the chances that the measured Δt will be anomalous and appear as an outlier.

Even given this type of integration strategy, there is a caveat. The TDC is subject to upset caused by glitches that occur along the core logic paths. Therefore, ATPG needs to be constrained to generate 'hazard-free robust' two-pattern delay tests. Although this restricts the number and types of tests that can be applied, it should be possible to obtain reasonably high test coverage across different regions of the chip, as recently demonstrated in [15].

The test generation (ATPG) strategy that we propose is based on the transition fault model commonly used by the manufacturing test community for delay faults. The transition fault model assumes a defect introduces a 'lumped' delay, as opposed to the path delay fault model which assumes a defect, and its corresponding delay, is distributed along the entire path. The lumped delay model is a better match for the capacitive loading and inserted gate delays introduced by Trojans.

There are several major benefits of the transition fault model. First, the lumped delay assumption allows ATPG to target tests for nodes, instead of paths, reducing the required number of tests to just 2 times the number of nodes, i.e., a rising and falling transition test for each node. Moreover, given that the number of paths is much larger (exponential to the number of nodes), each node can be tested along multiple different paths, making it more difficult for adversaries to 'hide' the additional delay by reducing delays in upstream and downstream gates. Last, given that the TDC provides precise delay information (not upper bounds as is true for conventional delay test methods), it can be used to measure delay variations on paths of *any* length. These latter two benefits increase the number of options available during the ATPG process, and work to both simplify and strengthen the test generation process.

7 CONCLUSIONS

Several statistical methods for detecting Trojans are proposed and evaluated on a set of FPGAs. Measurement noise, as well as within-die and chip-to-chip process variations, are analyzed to determine the impact of these noise sources on statistical limits. A regional calibration method is proposed that is capable of significantly increasing the level of sensitivity of the statistical methods to small current and frequency anomalies introduced by Trojans. High resolution power and delay methods are proposed to meet the sensitivity requirements for detecting subtle Trojan-induced variations in these circuit parameters.

8 REFERENCES

[1] K. Katsuki, M. Kotani, K. Kobayashi, and H. Onodera, "Measurement Results of Within-Die Variations on a 90nm LUT Array for Speed and Yield Enhancement of Reconfigurable Devices," in *Proc. Asia South Pacific Design Automation Conference*, pp. 110-111, 2006.

[2] P. Sedcole and P. Y. K. Cheung, "Within-Die Delay Variability in 90nm FPGAs and Beyond," in *Proc. International Conference on Field Programmable Technology*, pp. 97-104, 2006.

[3] B. Hargreaves, H. Hult, and S. Reda, "Within-Die Process Variations: How Accurately Can They Be Statistically Modeled?," in *Proc. Asia and South Pacific Design Automation Conference*, pp. 524-530, 2008.

[4] A. Maiti, P. Schaumont "Improving the Quality of a Physical Unclonable Function using Configurable Ring Oscillators," in *Proc. International Conference on Field Programmable Logic and Applications*, pp. 703- 707, 2009.

[5] Y. Jin, N. Kupp, and Y. Makris , "Experiences in Hardware Trojan Design and Implementation," in *Proc. International Workshop on Hardware-Oriented Security and Trust*, pp.

50-57, 2009.

[6] Y. Jin; Y. Makris, "Hardware Trojan Detection using Path Delay Fingerprint," in *Proc. International Workshop on Hardware-Oriented Security and Trust*, pp. 51-57, 2008.

[7] L. Jie, J. Lach, "At-Speed Delay Characterization for IC Authentication and Trojan Horse Detection," in *Proc. Workshop on Hardware-Oriented Security and Trust*, pp. 8-14, 2008.

[8] S. Narasimhan, D. Du, R. S. Chakraborty, S. Paul, F. Wolff, C. Papachristou, K. Roy, S. Bhunia, "Multiple-Parameter Side-Channel Analysis: A Non-Invasive Hardware Trojan Detection Approach," in *Proc. International Symposium on Hardware-Oriented Security and Trust,* pp. 13-18, 2010.

[9] D. Du, S. Narasimhan, R. S. Chakraborthy, S. Bhunia, "Self-Referencing: A Scalable Side-Channel Approach for Hardware Trojans Detection," *CHES*, 2010.

[10] R. M. Rad, X. Wang, M. Tehranipoor, J. Plusquellic, "Power Supply Signal Calibration Techniques for Improving Detection Resolution to Hardware Trojans," in *Proc. International Conference on Computer-Aided Design*, pp. 632-639*,* 2008.

[11] J. Aarestad, D. Acharyya, R. M. Rad and J. Plusquellic, "Detecting Trojans Though Leakage Current Analysis Using Multiple Supply Pad I_{DDQ}s," *Transactions on Information Forensics and Security*, pp. 893-904, 2010.

[12] H. Salmani, M. Tehranipoor, J. Plusquellic, "A Layout-aware Approach for Improving Localized Switching to Detect Hardware Trojans in Integrated Circuits," accepted *International Workshop on Information Forensics and Security*, 2010.

[13] D. Rai, J. Lach, "Performance of Delay-Based Trojan Detection Techniques under Parameter Variations," in *Proc. International Symposium on Hardware-Oriented Security and Trust,* pp. 58-65, 2009.

[14] J. Kalisz, "Review of Methods for Time Interval Measurements with Picosecond Resolution", *Metrologia*, 41, pp. 17-32, 2003.

[15] S. Menon, A.D. Singh, V. Agrawal, "Output Hazard-Free Transition Delay Fault Test Generation," in *Proc VLSI Test Symposium*, pp. 97-102, 2009.