Trojan Detection based on Delay Variations Measured using a High-Precision, Low-Overhead Embedded Test Structure

Charles Lamech, Jim Plusquellic
ECE Department, University of New Mexico, Albuquerque, NM
(clamech, jimp) @ece.unm.edu

Abstract - **The horizontal dissemination of the chip fabrication industry has raised new concerns over Integrated Circuit (IC) Trust, in particular, the threat of malicious functionality, i.e., a Hardware Trojan, that is added by an adversary to an IC. In this paper, we propose the use of a high-precision, low-overhead embedded test structure for measuring path delays to detect the delay anomalies introduced by hardware Trojans. The proposed test structure, called REBEL, is minimally invasive to the design as it leverages the existing scan structures. In this work, we integrate REBEL into a structural description of a pipelined Floating Point Unit. Trojan emulation circuits, designed to model internal wire loads introduced by a hardware Trojan, are inserted into the design at multiple places. The emulation cell incorporates an analog control pin to allow a variety of hardware Trojan loading scenarios to be investigated. We evaluate the detection sensitivity of REBEL for detecting hardware Trojans using regression analysis and hardware data collected from 62 copies of the chip fabricated in 90nm CMOS technology.**

***Keywords** - Hardware Trojans, Path Delay, Embedded Test Structure, Regression Analysis.*

I Introduction

A Hardware Trojan (HT) is defined as a deliberate and malicious modification to the functionality of the integrated circuit (IC) introduced by an adversary. Such modifications are designed to leak confidential information covertly or shut down the chip at some pre-determined time and/or when a specific data pattern is received. Many types of hardware systems from military applications to household appliances are threatened by HTs. Various detection strategies have been proposed in the past and they fall into three major categories: 1) IC de-processing and failure analysis, 2) functional activation through logic testing and 3) parametric anomaly detection. Of these approaches, parametric anomaly detection is more attractive for several reasons. First, the measurement of parametric signals can be done in a minimally invasive manner and is non-destructive. Second, the analog nature of the signal measurements provides high resolution to signal anomalies introduced by HTs. Last, calibration and statistical techniques can be used to eliminate and/or account for process variation effects, which can significantly improve the ability of such techniques to correctly identify HTs.

In this paper, we propose a high-precision, low-overhead embedded test structure (ETS) for detecting delay anomalies introduced by HTs. The proposed ETS, called REBEL for REgional dELay Behavior, is capable of delivering high resolution measurements of path delays and therefore is able to detect a wide range of delay anomalies introduced by HTs. REBEL is minimally invasive as it leverages the scan structures that already exist in the designs. In this paper, we investigate the detection sensitivity of REBEL using hardware experiments carried out on an ASIC.

The control logic integrated into REBEL allows for both the creation of a delay chain in a segment of scan chain and the selection of a path-under-test (PUT) to drive its input [1]. This is achieved by modifying the logic of the scan control path in the design. A standard launch-off-capture transition test vector is used to create the transition in the PUT which propagates through the core logic and then along this delay chain. The propagation is stopped after a specific period, called the launch-capture (LC) period, which takes a 'digital snap-shot' of the propagating signal, effectively digitizing the voltage behavior of the path's output. The delay of the PUT is calculated by subtracting the delay along the delay chain from the LC period.

REBEL provides several significant benefits over traditional delay testing methods. First, measuring the delays of short paths is difficult using traditional methods because the clock has to be run at speeds that exceed the operational frequency of the chip. REBEL, on the other hand, allows the use of slower-than-at-speed clocks for these paths by extending them into a delay chain. Second, the digital snap-shot captured by REBEL allows glitches to be detected, and therefore path delay measurements can be easily validated. Last, REBEL can potentially speed up the path delay measurement process because it captures the temporal behavior of paths in a sequence of flip-flops. Therefore, path delay measurements can be obtained using a small number of repeated application of the test pattern, with as few as only one depending on the desired timing accuracy. In contrast, clock strobing techniques require many applications of the test pattern sequence to achieve the same result.

These capabilities of REBEL allow it to be used in several contexts including defect detection, design-for-manufacturability, design debug and hardware security. In this paper, our focus is on using REBEL to detect HTs. HTs impact delay in two ways: either by adding capacitive load to existing wires in the design or by the insertion of additional gates in series with those of the original design. We designed a Trojan emulation circuit to model either of these two scenarios, and inserted it into multiple paths of a floating point unit (FPU). The Trojan emulation circuit incorporates an analog control pin to allow control over the amount of delay that is added to a PUT.

The Trojan emulation circuit is designed so that it can be disabled, allowing the Trojan-free behavior of these paths to be derived from the actual hardware (in typical applications, the 'golden' model would be derived from simulation experiments). The delays measured in the Trojan-free experiments are used to define statistical limits, which establish the boundaries against which the delays measured using the emulated HTs are compared. The statistical limits account for chip-to-chip process variations. A statistical technique called regression is used to classify a delay as 'normal' or 'anomalous'. We investigate the detection sensitivity of REBEL by varying the analog control voltage on each of Trojan emulation circuits (one at a time) in a sequence of experiments, and classify the resulting delay as normal or anomalous.

The remainder of this paper is organized as follows: Section II, discusses related works. In Section III, we describe the architecture and working principle of the proposed embedded test structure. Section IV explains the experimental design and setup. In Section V, we present the experimental results. Finally, conclusions are drawn in Section VI.

II Background

Several parametric-based approaches for detecting HTs have been proposed including methods based on delay, power, temperature, electromagnetic profile and leakage [2][3][4][5]. The authors of [3] propose a method that measures power transients and constructs a fingerprint of each chip. The fingerprints are tested against statistical limits derived using a small set of 'golden' chips that are later destructively validated. A second power transient technique is proposed in [2] for HTs that uses signal calibration, multiple supply ports and applies frequency domain analysis. The authors of [5] propose a HT detection strategy based on the leakage current analysis. The authors in [4] propose a statistical technique for detecting HTs that fingerprints ICs based on path delay measurements. In [6], the authors propose a gate-level characterization as a detection method. The authors of [7] use multiple side-channel parameters, in particular power and delay, as a means of improving HT detection sensitivity.

Several on-chip schemes are also proposed that facilitate HT detection. The authors in [8] propose a shadow register insertion and clock strobing technique designed to improve the on-chip path delay measurement accuracy. In [9], the authors propose a ring oscillator network scheme that has the ability to detect the power fluctuations caused by the malicious inclusions. Our approach is distinguished from these proposed methods primarily by the ETS that we use to measure path delays, and by its timing and area efficiency.

In [10], the authors show that the delay anomalies introduced by 'trigger Trojans', i.e., HTs that add capacitive load as fan-out, are subtle and harder to detect than HTs inserted in series with gates along a path. Our experiments are designed to investigate both types of HT insertions, but heavy focus is placed on analyzing the detection sensitivity of our method to the more difficult to detect 'trigger Trojans' types.

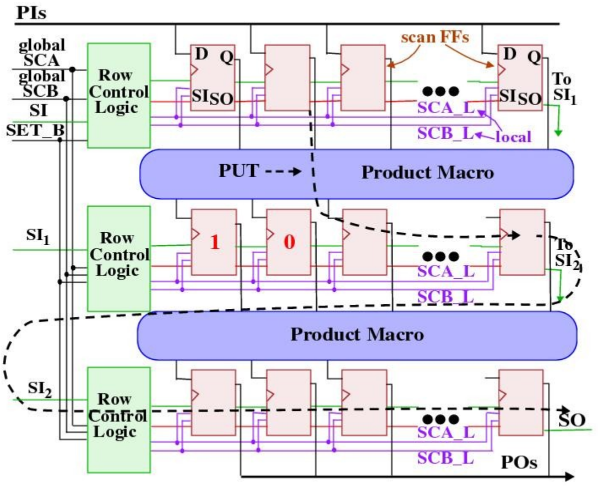

Fig. 1: REBEL Integration with CLSSD based Scan Chain

III Embedded Test Structure Design

A. REBEL Test Structure

In this section, we describe the design and working principle of the REBEL embedded test structure (ETS). REBEL exploits the scan chain by adding logic to the scan control path to enable specific segments of the scan chain to be converted into a delay chain. The additional logic also allows the output of the PUT to drive the input of the delay chain. A transition is created in the PUT and allowed to propagate along the delay chain for a specific launch-capture (LC) time interval. The PUT's delay is calculated by subtracting the delay chain component from the LC time interval. The delay chain component is measured in a separate calibration process.

In order to allow for the creation of a delay chain in a specific segment of the scan chain, the scan chain has to be partitioned into different segments. A Row Control Logic (RCL) block is added in front of each segment which controls the mode of operation of that segment. Fig. 1 shows the modifications required to incorporate REBEL into a level-sensitive-scan-design (LSSD) style scan chain. Here the scan chain is partitioned into three segments (shown as three rows in Fig. 1). The RCL block, shown on the left in the figure, controls the operational mode of the row. For example, the top row is configured to operate in functional mode, which allows a transition test to be applied to the core logic. The middle row shows the insertion point of the PUT's output into the delay chain. The scan FFs to the right of this insertion point are configured into 'flush-delay' (FD) mode to allow the PUT's output signal to propagate along the delay chain. The last row simply extends the length of the delay chain. Reference [1] give the details of the RCL block and describes the scan chain modifications necessary to allow these modes of operation.

A transition is created in the PUT using traditional launch-off-capture transition fault test. In this scenario, the scan chain is loaded with the initial pattern of the transition test and the system clock (CLK) is used to generate a transition in the core logic by capturing the output of a previous block, or by capturing the primary input (PI) values, as shown in Fig. 1. The transition emerges on an output of the macro, and drives the input of

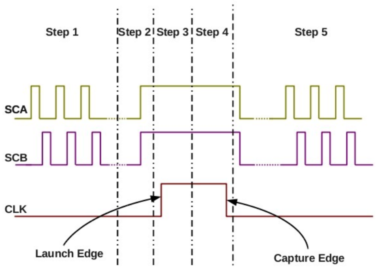

Fig. 2: REBEL Test Timing Diagram

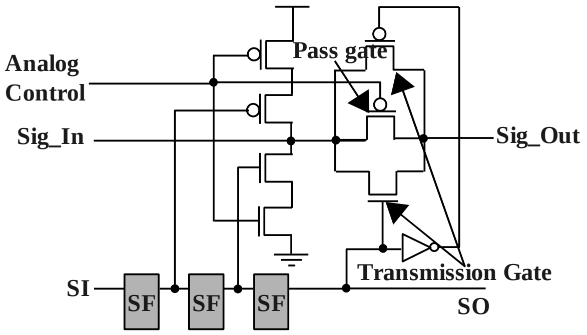

Fig. 3: Trojan Emulation Circuit

the delay chain. The propagation of the edge is halted by de-asserting the CLK after the LC time interval, which effectively takes a 'digital snap-shot' of the signal propagation behavior along the delay chain, including any glitching that may have occurred. This digital snap-shot is then scanned out for analysis.

The delay along the delay chain is determined by a separate process called calibration. Calibration involves placing all of the scan cells in the delay chain in flush-delay mode, and then launching an edge into the SI pin of the scan chain. A sequence of LC tests are carried out using incrementally longer LC time intervals.

The discrete nature of the delay chain quantizes the analog delay into intervals given by the inter-scan-FF delay, i.e., the time taken for a edge to propagate from one scan FF to the next. This inter-scan-FF delay is technology dependent but is typically larger than the desired timing resolution (for example, it is approx. 500 ps in our experiments). One way to improve timing resolution is by *clock strobing*, which involves repeating the transition test using much smaller timing increments, e.g., 25 ps. Clock strobing allows a high resolution delay map of the delay chain to be constructed that reflects the precise delay of the individual components of the delay chain. The same process can be applied to obtain the delay of the PUT at high resolution.

B. *Path Delay Measurement Procedure*

The LC period is controlled by the system clock (CLK) and therefore REBEL leverages the clock tree for critical timing events. Fig. 2 shows the timing diagram of the REBEL test which is carried out as follows:

1. The initial test vector and configuration data are scanned in.
2. SCA and SCB are asserted to move into FD mode and establish the delay chain.
3. The CLK is asserted to launch a transition into the PUT.
4. The CLK is de-asserted after a specific Δt, sufficiently long enough to allow the transition to enter and propagate along the delay chain.
5. SCA and SCB are de-asserted and the data is unloaded using scan for analysis.

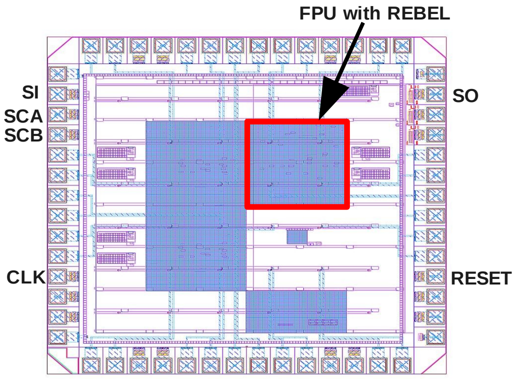

Fig. 4: Chip Top Level Layout

The PUT's delay is computed by using Eq. 1

$$T_{path} = T_{lc} - T_{sc} \qquad \text{Eqn. 1}$$

where, T_{path} = Delay in the PUT
T_{lc} = Launch/Capture time period
T_{sc} = Delay along the delay chain

IVExperimental Design and Setup

A. *Trojan Emulation Circuit*

The Trojan emulation circuit is designed to introduce delay anomalies in the path that model the additional capacitive loading imposed by Trojan circuits. Fig. 3 shows the internal structure of the Trojan emulation circuit. The operation of the Trojan emulation circuit is controlled by three configuration registers. Setting the configuration registers to '101' disables the Trojan emulation circuit and represents the Trojan-free design. In this scenario, both the NMOS and PMOS transistors of the transmission gate are enabled. In order to emulate a Trojan, the transmission gate is disabled by setting the state of the configuration registers to '100'. With the transmission gate disabled, a resistive connection can be configured between Sig_In and Sig_Out by tuning the voltage on the analog control pin to a value above 0 V. The analog control pin connects to the gate of the PMOS pass gate, so values above 0 V leave it only partially conducting. For example, setting the analog control pin to 0.6 V (one half of the supply voltage) will introduce a delay anomaly for a rising or falling edge that propagates though the cell. The stacked PMOS and NMOS transistors in the left portion of the Trojan emulation cell can also be used in a similar fashion to create delay anomalies by setting the state of the configuration resisters to '111'.

B. Floating Point Unit with REBEL structure

REBEL is integrated with a pipelined Floating Point Unit (FPU) which has a scan chain length of 671 bits. The scan chain is divided into 28 segments as a strategy to enhance the coverage achievable for path delay Automatic Test Pattern Generation (ATPG). An RCL block is added to control each of these segments as explained in Section IIIA. In addition, multiple copies of the Trojan emulation circuit are added to the verilog structural description of the FPU. The structural description is synthesized to a layout using the Cadence Encounter Place and Route tool. The final design is fabricated in IBM's 90nm CMOS bulk process. Fig. 4 shows the top level layout of the chip and the region in which the FPU is integrated. We recently received and tested 62 copies of the chip.

TABLE I. Area Overhead of REBEL

Macro	Area(μm^2)	No of Scan Cells	No of Rows	Area Overhead
FPU	251763.09	761	28	11.45%
AES	251763.09	530	18	7.7%

Table I give the area overhead associated with REBEL with respect to the logic needed to implement the FPU and in a second AES (Advanced Encryption Standard) macro that is also included on the chip. The area overhead of REBEL depends on: 1) the amount of sequential logic in the design, and 2) the number of segments the scan chain is divided into. From the table, it is clear that the amount of sequential logic and the corresponding number of segments is larger in FPU than in AES which results in higher overhead. A closed form expression for estimating the amount of overhead associated with REBEL is given as follows:

$$REBEL_{Overhead} = N_{sc} * A_{CTRL} + N_{seg} * A_{RCL} \quad \text{Eqn. 2}$$

where, N_{SC} = Number of Scan Cells,
A_{CTRL} = Area of additional logic added to scan cell,
N_{seg} = Number of segments,
A_{RCL} = Area of Row Control Logic.

The first term in Eqn. 2 is the area overhead introduced by the modifications required to the scan cells. The second term in the Eqn. 2 is the area overhead introduced by the RCL logic which depends on the number of segments that are created in the scan chain. The overhead associated with the RCL blocks can be reduced at the expense of reducing path coverage.

C. Experimental Setup

Fig. 5 shows the instrumentation that we used for the REBEL tests. An Inovys structural tester, donated by Verigy Inc., is used to apply the test vectors and to capture the responses of the delay tests. High speed differential clocks generated in the High Performance Clock Channels (HPCC) of the tester are used to generate precisely timed launch/capture edges. An on board line receiver is used to convert the differential signal into a single ended clock at a point very close to the zero-insertion-force socket for the chip. The transition fault test vectors used in our experiments are created using the Cadence Encounter Test ATPG tool.

V Experimental Results

In this section, we present the results of applying our technique to the FPU macro on a set of 62 chips. The first step of the process involves applying a set of calibration tests to the FPU scan chain as a means of characterizing delays along its segments. The calibration data is used to compute the actual path delays from a subsequent set of transition tests applied to the core logic of the FPU. We apply two types of transition tests to the FPU, one set in which all Trojan emulation circuits are disabled and a second set that enables exactly one Trojan emulation circuit at a time. The delays measured under the Trojan-free tests are used to derive statistical limits. These limits are later used to classify the delays measured under the HT tests.

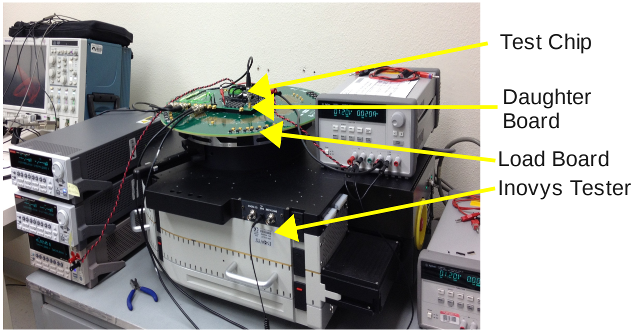

Fig. 5: Experimental Setup

A. Calibration

As mentioned in section II.A, the calibration process for REBEL is designed to eliminate delay of the PUT's transition along the delay chain, as specified by T_{sc} in Eqn. 1. From preliminary experiments, we determined that the uncertainty in the delays measured in the calibration process grows as the edge propagates further along the delay chain. This uncertainty increases the measurement error in the calculated path delays. In order to minimize these measurement errors, we created a special set of transitions in the combinational logic for calibration. These special logic tests are designed to drive a transition into the first (left-most) flip-flop of each segment. This technique reduces the uncertainty in the delays by introducing a set of transitions that are derived from nearby logic. The different velocities associated with the propagation of a rising and falling edge in the scan chain require both rising and falling edge calibration tests.

In our experiments, we set the LC resolution to 25 ps. Once the calibration tests are performed, the delay along any portion of the delay chain can be computed as given by Eqn. 3. The expression gives the delay along the delay chain (T_{SC}) between two scan flip-flops FFe and FFs as the difference in the delays measured under the calibration tests to these scan FFs, i.e., T_{ffe} and T_{ffs}.

$$T_{sc} = T_{ffe} - T_{ffs} \quad \text{Eqn. 3}$$

where, T_{sc} = Delay along the delay chain
T_{ffe} = LC period for the calibration edge to reach 'FFe'.
T_{ffs} = LC period for the calibration edge to reach 'FFs'.

The calibration process needs to be carried out first, and at the same resolution that is used for the logic tests. The time required to carry out calibration depends on the length of the delay chain and the desired resolution, but in any case, it represents overhead that we would like to minimize. In the analysis of our chip data, we found that the calibration curves are similar in shape across the chips and that most of the differences between them can be eliminated by multiplying them by a scaling factor. In other words, regional, within-chip variations are insignificant compared to the chip-to-chip variations in this technology. We realize that this may not hold true in more advanced technologies, and calibration may be needed for every chip. However, in cases where scaling is possible, this short-cut can be used to significantly reduce overhead.

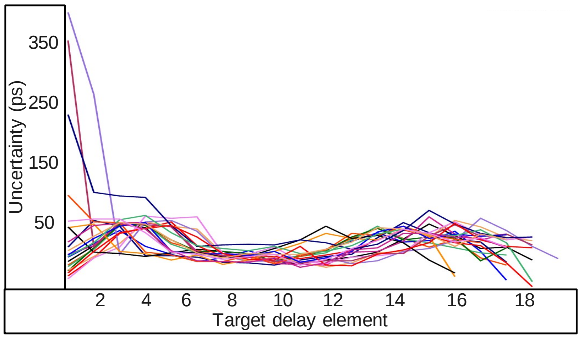

Fig. 6:Uncertainty at Various Target Flip-Flops

We also investigate the elimination of the calibration process altogether. Technically, we do not need to know the actual path delays in order to detect anomalies introduced by HTs. However, including the delay along the delay chain in the statistical detection method (described below) acts to 'water-down' the anomaly introduced by the HT, and reduces detection sensitivity. We refer to 'uncalibrated' delays as delays that include the delay chain component in the following sections.

B. *Noise Analysis*

As described earlier, clock strobing is applied using incrementally longer LC intervals to determine the delays between elements of the delay chain (referred to as inter-scan-FF delays earlier). The LC interval in which a delay element is first able to record the transition is the interval used to determine the inter-scan-FF delay. However, this LC interval may vary from one sample to the next. The range over which it varies is referred to as *uncertainty*. Uncertainty reflects the noise level present in the measurements.

In our experiments, we found that the level of uncertainty varies for delay elements at different positions from the PUT's insertion point. Given that REBEL allows a 'target' delay element to be chosen, this characteristic of uncertainty can be leveraged as a means of improving resolution. Fig. 6 plots uncertainties along the y-axis for each of the numbered delay elements along the x-axis. The delay elements beginning from the PUT's insertion point (the capture FF) are numbered 1 to 18 for each of the consecutive elements along the delay chain. Each of the curves represents the set of uncertainties associated with one PUT. The graph plots the results for 22 PUTs.

The wide range of uncertainties on the left side of the graph indicate that delay elements near the insertion point work well as a target for some PUTs but very poorly for others. This occurs because the voltage behavior on the outputs of some PUTs glitch, making it difficult to obtain a consistent measurement. On the other hand, the uncertainties for target delay elements in the center region of the graph, e.g., delay elements numbered 8 and 9, provide the lowest overall uncertainty across the PUTs. Uncertainties again increase for delay elements at larger distances from the insertion point because of power supply noise, signal coupling and other types of on-chip noise. Given these characteristics, we use the 9th delay element to compute path delays.

The average uncertainty computed from all path delay measurements in our experiments is approx. 40 ps, and the worst case uncertainty is approx. 150 ps. The main sources of this uncertainty are meta-stability in the delay elements, power supply noise, clock glitter and temperature variations. The worst case uncertainty may be smaller for designs that generate the LC timing events on-chip.

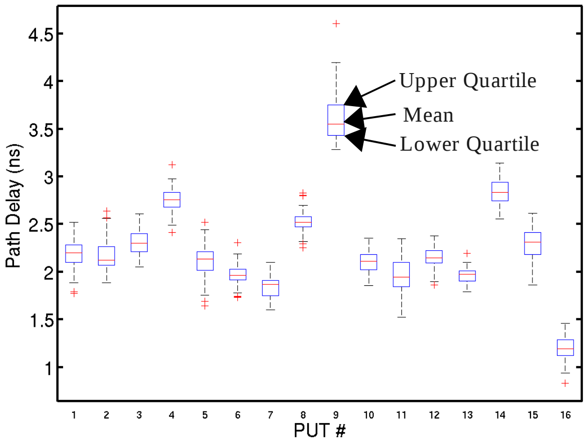

Fig. 7:Noise and Process Variation: Box Plot of 16 Different Path Delays among 62 chips

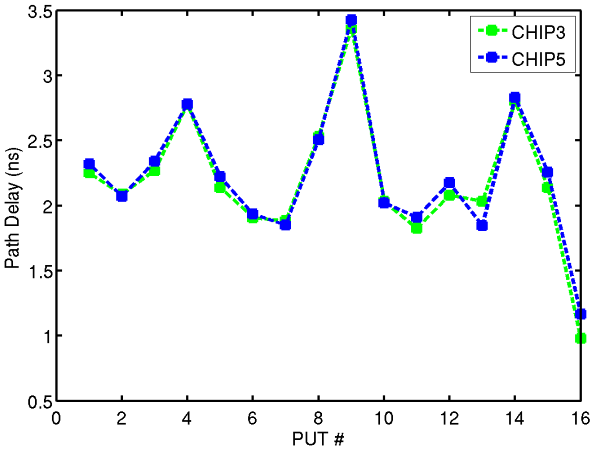

Fig. 8:Within-Chip Variation: CHIP3 and CHIP5

C. *Analysis of Process Variations*

The path delays from a set of Trojan-free transition tests are used to analyze delay variations caused by within-chip and chip-to-chip process variations. Fig. 7 gives a boxplot analysis of PUT delays from 62 chips and 16 transition tests. The figure plots the PUT number on the x-axis against the measured delays on the y-axis. The boxplot associated with each PUT identifies the median, upper and lower quartiles as horizontal lines. A set of dotted lines extend to the largest and smallest delays and a '+' indicator identifies outliers in the distribution. As expected, chip-to-chip variation increases as the length of the path increases, e.g., the variation portrayed by the boxplot for PUT #9 is larger than that for PUT #10.

The curves shown in Fig. 8 plot PUT # against delay in the same manner as in Fig. 7, except this time, only 2 chips are shown and the delays for a given chip are connected together in a curve. The close match of the two curves indicates that these two chips have similar overall performance characteristics. Crossings that occur between the curves illustrate cases where within-chip variation and/or measurement noise changes the relative delay values of the PUTs. Although this occurs frequently in the

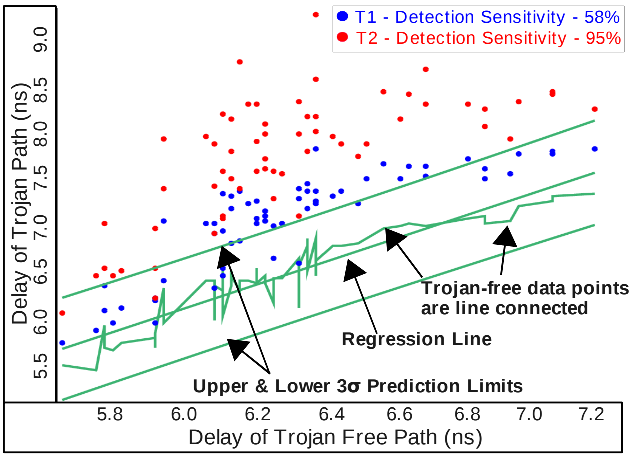

Fig. 9: Regression Analysis using uncalibrated delays – 62 Trojan-Free data points and 62 Trojan data points for each of the 2 Trojans

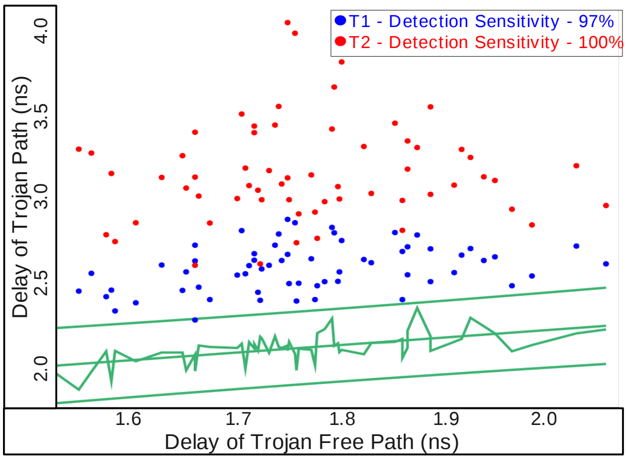

Fig. 10: Regression analysis calibrated delays - 62 Trojan-Free data points and 62 Trojan data points for each of the 2 Trojans

curves, the magnitudes of the vertical deviations are very small, indicating that within-die variations and noise are insignificant relative to the chip-to-chip variations depicted in Fig. 7. The statistical technique that we describe in the next section can deal effectively with chip-to-chip variations. However, within-chip and variations and noise act to reduce its sensitivity. The fact that they are small in this technology allows very small delay anomalies to be detected.

D. *Emulated Trojan Characteristics*

As indicated earlier, delay anomalies are created using HT emulation circuits, which are designed to model the capacitive loads HTs introduce on existing wires in the design. The magnitude of the delay anomaly is controlled using an analog voltage pin that connects to a PMOS passgate. Analog voltages above 0 V increase the on-resistance (R_{ONP}) of the PMOS passgate and increase delay according to Eqn. 4:

$$\Delta T \approx 0.7 * \Delta R_{ONP} * C_{Load} \qquad \text{Eqn. 4}$$

where, ΔR_{ONP} is the change in the PMOS on-resistance due to change in analog voltage, and C_{Load} is the load capacitance.

TABLE II. Analog Voltage Change and Corresponding Additional Load Capacitance

Analog Voltage Change (ΔVg)(V)	ΔR_{ONP} (kΩ) (appx.)	ΔC_L (fF) (appx.)
0.1	1.4	1.0
0.2	2.1	3.0
0.3	3.0	5.0
0.4	4.5	7.0
0.5	7.0	10.0
0.6	18.0	17.0
0.7	162.0	25.0

We solve this equation using simulation experiments carried out on a extracted netlist representing a 'nominal' model of the FPU. The results shown in Table II give the additional resistance added to the path for the set of analog voltages listed in the left-most column. An approximation of the equivalent additional load capacitances can be obtained by plugging in the ΔR_{ONP} values from the table into Eqn. 4. The results are given in right-most column of Table II. The small values of ΔC_L in the top-most rows indicate that the Trojan emulation circuit is capable of introducing very subtle changes in path delays.

E. Calibrated vs. Uncalibrated Delays

We indicated earlier that calibrating path delays, i.e., removing the delay chain component, is optional but can improve detection sensitivity. Figs. 9 and 10 are used to illustrate this point using regression analysis. Regression analysis operates on scatterplots where the delays from one PUT are plotted against the delays from a second PUT.

The scatterplots of Figs. 9 and 10 plot the delays measured from two paths in the chips. The path plotted along the x-axis is a Trojan-free path while the path plotted along the y-axis includes an emulated HT circuit. Fig. 9 uses delays that are a sum of the delays from the PUT and delay chain ('uncalibrated') while Fig. 10 uses delays with the delay chain component subtracted (calibrated). Regression lines or 'best-fit' lines are derived using the Trojan-free data points in both figures. The delays representing the path plotted along the y-axis in this case are measured with the emulated Trojan circuit disabled. The curves labeled "3 σ prediction limits" represent the expected variation in these paths. The expected variation accounts for within-chip process variations and noise. The delays from Trojan-free chips are expected to fall within the region delineated by these curves. Chip-to-chip process variations, on the other hand, are tracked along the length of the regression line and therefore, are effectively eliminated as detractors to HT detection sensitivity.

The plots of Figs. 9 and 10 also include data points with the emulated Trojan circuit activated using two different analog voltages (labeled as T1 and T2). The analog voltages used are equivalent to load capacitances of 10 fF (T1) and 17 fF (T2). The HT data points are depicted as shaded points in the figures, with the darker shaded points corresponding to T1. The impact of including the delay chain component can be evaluated by counting the number of HT data points that fall outside of the prediction limits in Figs. 9 and 10. The number of detections is larger in Fig. 10 than in Fig. 9, indicating that removing the delay chain component improves detection sensitivity. In particular, 97% of the T1 data points and 100% of the T2 data points are detected in Fig. 10, while only 58% and 95% are detected in Fig. 9.

Calibrated data improves sensitivity because the anomaly introduced by the HT is relatively larger with the delay chain component eliminated. Moreover, any delay variations that occurs within the delay chain due to within-chip process variations are also eliminated. The drawback of using calibrated delays is that the uncertainty in the measurements is effectively doubled because two measurements are used. For example, in our experiments, the single measurement uncertainty of 150 ps doubles to 300 ps.

F. Hardware Trojan Results

Path delay ATPG is used to derive tests for 6 of the embedded HT emulation circuits. Delays measured from these paths, with and without the HT enabled, are used in regression analysis, and detection statistics are computed using the set of test chips.

The results are summarized in Table III for each of the HTs identified by path names TP1 through TP6. The second column gives the average delay of the path across the 62 chips. The remaining columns give the number of positive detections as a percentage. The percentages are computed by counting the number of data points that fall outside the 3 σ prediction limits in each of the six scatterplots. These numbers are then divided by the number of chips and multiplied by 100. Each emulated HT is tested at 7 different analog voltages. The analog voltages are translated into equivalent additional load capacitances using Eqn. 4, and are given in the header row of the table.

TABLE III. Trojan Detectability For Various Trojans

Trojan Detection Percentage (%)								
Path Name	**Path Delay (ns)**	**ΔC_L (fF)**						
		1	**3**	**5**	**7**	**10**	**17**	**25**
TP1	2.175	0	4	10	52	97	100	100
TP2	1.900	0	0	2	8	40	94	100
TP3	4.925	51	100	100	100	100	100	100
TP4	3.500	4	42	93	100	100	100	100
TP5	1.200	0	0	7	48	86	100	100
TP6	1.575	0	1	19	77	100	100	100

As expected, smaller capacitive loads are harder to detect. For example, a HT modeled as adding 7 fF to a node along the path named TP1 is detected approx. half of the time (52%), but increases to 97% when the additional capacitance increases to 10 fF.

G. Sensitivity Analysis

From Table III, a relationship appears to exist between the smallest capacitive load that first produces positive detections and path length. For example, two of the shortest paths, TP5 and TP2, are not able to detect anomalies until the capacitive loads reach 5 fF while the longest paths, TP4 and TP3, are able to detect capacitive loads as small as 1 fF. This at first seems counter-intuitive because one would expect that short paths would be changed more dramatically by smaller capacitive loads than longer paths, under the condition that the delta increase in delay introduced by the capacitive load is constant and independent of path length. For example, if the capacitive load of a HT increases the delay of a path by 1 ns, then the delay of a short path with nominal delay of 2 ns is increased by 50%, whereas the delay of a longer path with nominal delay 4 ns is increased by only 25%.

The reason the opposite appears to hold in our data is rooted in the "averaging" effect that occurs for longer paths. Random with-chip variations in the gate delays of longer paths tend to average out and exhibit lower levels of overall path delay variation in the chip population. This is reflected by the position of the statistical limits in our analysis. In particular, we found that the distance of the statistical limits from the regression lines for longer paths is smaller than it is for shorter paths.

H. Test Time Analysis

In this work, we test only a small number of paths in the FPU. In any practical application of this technique, a much larger number of paths would need to be tested to gain sufficient confidence that the chips are free of HTs. Test time then becomes an important issue to consider because it relates to the cost effectiveness of the technique. The clock strobing technique discussed in the previous section can be very expensive in terms of test time if it is applied in a step-wise linear fashion. The temporal visibility that REBEL provides allows a much faster search process to be carried out to find the LC interval that achieves the goal of propagating the edge to a target delay element (as we discussed in Section III). The LC interval can be 'intelligently tuned' based on the results of the previous application of the test sequence, which can dramatically reduce the number of repeated applications of the test sequence.

A second cost cutting measure is to eliminate the calibration process. However, as we discussed in the previous section, calibration improves detection sensitivity. Therefore, a trade-off exists in this case. Bear in mind that the HT detection problem, unlike the defect detection problem, can leverage parallelism by applying different subsets of the test sequences to separate chips simultaneously. This can be very effective at reducing test costs but only works if we can assume that every chip is identical, i.e., the HT is not selectively inserted into only a subset of the chip population.

VI Conclusion

In this paper, we proposed the use of a high-precision, low overhead embedded test structure, called REBEL, for hardware Trojan detection. REBEL is integrated into a pipelined Floating Point Unit and implemented using design automation tools. The design overhead is small and integrating REBEL is easy because it can be automated completely within the DFT synthesis flow. Trojan emulation circuits are used to model the delay anomalies introduced by hardware Trojans. Path delays are measured in 62 copies of the chip fabricated in IBM's 90nm CMOS technology. Linear regression analysis is applied to the data sets and is shown to reduce chip-to-chip process variation effects and significantly improve detection sensitivity. The results show that REBEL can detect delay anomalies introduced by subtle changes in the capacitive loads introduced by HTs.

Acknowledgments

This work is supported by NSF grant CCF: 1118025. The 90nm chips are fabricated under MOSIS MEP program. We thank Dhruva Acharya and Erik Volkerink at Verigy Inc. and Kanak Agarwal and Sani Nassif at IBM Austin Research Laboratory for their support in this project.

References

[1] C. Lamech, et al., "REBEL and TDC: Two Embedded Test Structures for On-Chip Measurements of Within-Die Path Delay Variations", *ICCAD*, pp. 170 - 177, 2011.

[2] R. M. Rad, et al., "Power Supply Signal Calibration Techniques for Improving Detection Resolution to Hardware Trojans", *Proc. ICCAD*, pp. 632-639, 2008.

[3] D. Agrawal, et al., "Trojan Detection using IC Fingerprinting", *Symposium on Security and Privacy*, pp. 296-310, 1996.

[4] Y. Jin, et al., "Hardware Trojan Detection using Path Delay Fingerprint", *Proc. International Workshop on Hardware-Oriented Security and Trust*, pp. 50-57, 2008.

[5] J. Aarestad, et al., "Detecting Trojans Though Leakage Current Analysis Using Multiple Supply Pad IDDQ",*Transactions on Information Forensics and Security*, pp. 893-904, 2010.

[6] M. Potkonjak. et al., "Hardware Trojan horse detection using gate-level characterization", *DAC '09*, pp. 688-693, 2009.

[7] S. Narasimhan, et al., "Multiple-Parameter Side-Channel Analysis: A Non-Invasive Hardware Trojan Detection Approach", *Proc. International Symposium on Hardware-Oriented Security and Trust*, pp. 13-18, 2010.

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

[9] Xuehui Zhanget, et al., "RON: An on-chip ring oscillator network for hardware Trojan detection", *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1-6, 2011.

[10] C. Lamech, et al., "An Experimental Analysis of Power and Delay Signal-to-Noise Requirements for Detecting Trojans and Methods for Achieving the Required Detection Sensitivities", *IEEE Transactions on Information Forensics and Security* , pp. 1170-1179, 2011