Self-referencing: A Scalable Side-Channel Approach for Hardware Trojan Detection

Dongdong Du, Seetharam Narasimhan,
Rajat Subhra Chakraborty, and Swarup Bhunia

Case Western Reserve University, Cleveland OH-44106, USA
sxn124@case.edu

Abstract. Malicious modification of integrated circuits (ICs) in untrusted foundry, referred to as "Hardware Trojan", has emerged as a serious security threat. While side-channel analysis has been reported as an effective approach to detect hardware Trojans, increasing process variations in nanoscale technologies pose a major challenge, since process noise can easily mask the Trojan effect on a measured side-channel parameter, such as supply current. Besides, existing side-channel approaches suffer from reduced Trojan detection sensitivity with increasing design size. In this paper, we propose a novel scalable side-channel approach, named *self-referencing*, along with associated *vector generation algorithm* to improve the Hardware Trojan detection sensitivity under large process variations. It compares transient current signature of one region of an IC with that of another, thereby nullifying the effect of process noise by exploiting spatial correlation across regions in terms of process variations. To amplify the Trojan effect on supply current, we propose a region-based vector generation approach, which divides a circuit-under-test (CUT) into several regions and for each region, finds the test vectors which induce maximum activity in that region, while minimizing the activity in other regions. We show that the proposed side-channel approach is scalable with respect to both amount of process variations and design size. The approach is validated with both simulation and measurement results using an FPGA-based test setup for large designs including a 32-bit DLX processor core ($\sim 10^5$ transistors). Results shows that our approach can find ultra-small (<0.01% area) Trojans under large process variations of up to ± 20% shift in transistor threshold voltage.

Keywords: hardware Trojan, side-channel analysis, self-referencing.

1 Introduction

Global economics dictates increasing out-sourcing of Integrated Circuit (IC) fabrication to off-shore facilities. Though cost-effective, out-sourcing brings up potential risks for an adversary to maliciously modify a circuit. Such malicious modifications are referred as *Hardware Trojans.* A typical Hardware Trojan would cause an IC to have altered functional behavior during operation in the field, potentially with disastrous consequences in safety-critical applications. The

S. Mangard and F.-X. Standaert (Eds.): CHES 2010, LNCS 6225, pp. 173–187, 2010.

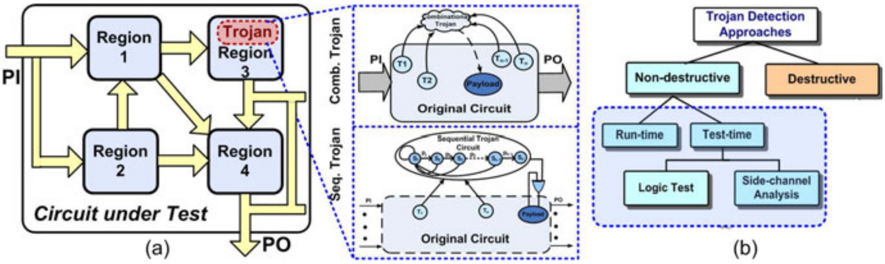

Fig. 1. (a) A circuit with hardware Trojan along with models of two types of Trojans. (b) A taxonomy of existing hardware Trojan detection techniques.

threat of Hardware Trojans has emerged as a major security concern [1], especially since several unexplained military mishaps are attributed to the presence of malicious hardware Trojans [2,3]. Such hardware Trojans can also be inserted in a design house during the design of an IC. Here, we focus on the problem of detecting hardware Trojans inserted during fabrication in an untrusted foundry.

An intelligent adversary can incorporate a hardware Trojan, which is extremely difficult to detect during conventional post-manufacturing test. Due to their stealthiness, Trojans can be triggered only under rare conditions. Upon triggering, they can either cause malfunction by altering internal node values [4] or leak secret information through covert channels [5]. They can also be used to assist software attacks by providing a hardware backdoor [3]. Fig. 1(a) shows an example circuit with Trojan inserted inside one of its constituent blocks. Broadly two types of Trojan can be inserted in a digital circuit: *combinational Trojans*, which are activated by a rare combination of values at internal circuit nodes and *sequential Trojans*, which are activated through a sequence of rare events. Several approaches to detect hardware Trojans have been proposed in recent literature [5]. We show a classification of the Trojan detection techniques in Fig. 1(b). Destructive testing of a chip by de-packaging, de-metallization and micro-photography based reverse-engineering is highly expensive (in time and cost) and not a feasible solution because an attacker may selectively insert Trojan into a small subset of the manufactured ICs [7]. Conventional logic testing, both functional and structural, performs poorly in detecting Trojans, due to their stealthiness, arbitrary nature and size [8]. An alternative approach is to measure a side-channel parameter, such as supply current or path delay, which can be affected due to unintended design modifications. However, the effectiveness of side-channel analysis is limited by large device parameter variations in modern nanometer technologies leading to variations in the measured side-channel parameter, which can mask the effect of a small Trojan.

The issue of process variations on side-channel analysis based Trojan detection has been considered in [9], which explores signal processing techniques to reduce effect of process noise on supply current. Another approach based on power-supply transient [6], measures current signal from multiple power ports and uses a statistical characterization of process noise. Path delays of output

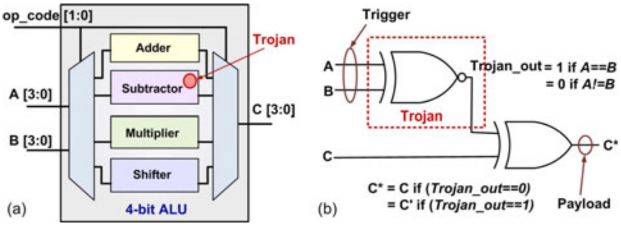

Fig. 2. (a) An simple test circuit: a 4-bit Arithmetic Logic Unit (ALU). (b) A combinational Trojan inserted into the subtractor.

ports have also been used as the fingerprint [11], with extensive characterization for process variations. In this paper, we propose a scalable side-channel approach to hardware Trojan detection based on a concept called "self-referencing". The basic idea is to use supply current signature of one region of a chip as reference to that of another to eliminate the process noise. Such calibration or referencing is possible due to the spatial correlation of process variation effects across regions in a chip. We show that such an approach can be extremely effective in nullifying all forms of process noise, namely inter-die, intra-die random and intra-die systematic variations [13]. Since process noise is eliminated by comparing current signature of regions in an IC, the method is scalable with increasing process noise, unlike existing approaches [9]. To increase the Trojan detection sensitivity, we propose a region-based vector generation approach, which tries to maximize the Trojan effect while minimizing the background current. Current values of n regions are then compared with all other using a slope heuristic and the resultant *region slope matrix* is used to compare a chip with another. We validate the proposed approach using both simulation and measurements for several large open source designs. Simulation results shows high detection sensitivity in presence of large process variations and scalability of the approach with increasing design size. The measurement results with a custom test board validates the effectiveness of the approach.

The rest of the paper is organized as follows. In section 2, we describe the motivation of the proposed self-referencing method. Section 3 presents the methodology along with theoretical analysis. Simulation and experimental results are described in section 4. Section 5 concludes the paper.

2 Motivation of Self-referencing Approach

The idea of self-referencing can be illustrated using an example 4-bit ALU, as shown in Fig. 2(a). The ALU contains four distinct functional units (FUs) - adder, subtractor, multiplier and shifter, which are activated based on the input "opcode" value. There are two 4-bit operands and a 4-bit output. In such a circuit, a single region or FU can be selectively activated by proper choice of opcode, we can easily generate test vectors which target separate activation of the four regions. We consider three different process corners (nominal ±25%) for

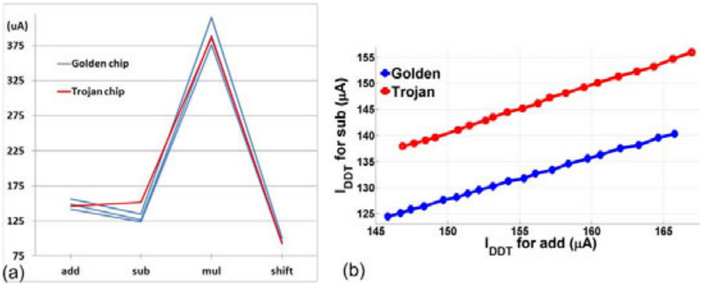

Fig. 3. (a) Comparison of supply current between golden and tampered chip for four regions of a 4-bit ALU. (b) Correlation of region currents at different process points for golden and tampered ICs.

the entire design (modeled as a change in the transistor threshold voltage V_T) and simulate the design in HSPICE for four different vector pairs which activate each of the four regions separately. We also measure the background current. The Trojan circuit, as shown in Fig. 2(b) was designed to invert an output bit of the subtractor if two input bits were equal. We simulated the circuit with the Trojan in the subtractor module (occupying 2.7% area of the ALU) at the nominal process corner for the same set of vectors.

Fig. 3(a) shows the plot of the average I_{DDT} values for the four different vectors activating the four different regions without the background current. We can observe the tampered circuit consumes more current for the vector which activates the subtractor region. We plot the current for one region (adder) with respect that for another (subtractor) for a set of golden and tampered chips at 20 different process points in Fig. 3(b). We expect a correlation between the region currents across process corners. However, since there is a Trojan in the subtractor, it shows uncorrelated behavior in supply current. Hence, the current for the adder can be used to calibrate the process noise and check for the presence of Trojan in other modules. In real life, since we do not know the region which contains the Trojan, we need to compare each region with all others. This also allows us to cancel out the effect of random and systematic intra-die process variations, as explained later.

3 Methodology

For a large design, the golden supply current for a high activity vector can be large compared to the additional current consumed by a small Trojan circuit, and the variation in the current value due to process variation can be very large. This can mask the effect of the Trojan on the measured current, leading to difficulty in detecting a Trojan-infected chip. Most side-channel analysis based approaches perform calibration of the process noise by using golden chips at different process corners. This helps us obtain a limiting threshold value beyond which any chip is classified as Trojan. Since the variation in the measured value can cause a golden chip to be misclassified as a Trojan (we refer to this case as a *false positive - FP*), the limit line has to be close to the nominal golden value. On the other hand, if the Trojan effect does not change the value beyond the limit, the Trojan-containing chip can be misclassified as a golden one (we refer to this case as a *false negative - FN*). To limit the probability of false positives and false negatives, the limiting values need to be chosen carefully.

The Trojan detection sensitivity of this approach reduces with decreasing Trojan or increasing circuit size. In order to detect small sequential/combinational Trojans in large circuits ($> 10^5$ transistors), we need to improve the SNR (Signal-to-Noise Ratio) using appropriate side-channel isolation techniques. At a single V_T point the sensitivity, for an approach where transient current values are compared for different chips, can be expressed as:

$$Sensitivity = \frac{I_{tampered,nominal} - I_{golden,nominal}}{I_{golden,process\ variation} - I_{golden,nominal}} \tag{1}$$

Clearly, the sensitivity can be improved by increasing the current contribution of the Trojan circuit relative to that of the original circuit. We can divide the original circuit into several small regions and measure the supply current (I_{DDT}) for each region. The relationship between region currents also helps to cancel the process variation effects. In Fig. 3(a), if we consider the “slope” or relative difference between the current values of ‘add’ and ‘sub’ regions, we can see that there is a larger shift in this value due to Trojan than in the original current value due to process variations. We refer to this approach as the *Self-Referencing* approach, since we can use the relative difference in the region current values to detect a Trojan by reducing the effect of process variations. In the appendix, we present an analysis regarding how the self-referencing approach can help cancel the effect of process variations.

The major steps of the self-referencing approach are as follows. First, we need to perform a *functional decomposition* to divide a large design into several small blocks or regions, so that we can activate them one region at a time. Next, we need a vector generation algorithm which can generate vectors that maximize the activity within one region while producing minimum activity in other regions. Also, the chosen set of test vectors should be capable of triggering most of the feasible Trojans in a given region. Then, we need to perform self-referencing among the measured supply current values. For this we use a *Region Slope Matrix* as described in the appendix. Finally, we reach the decision making process which is to compare the matrix values for the test chip to threshold values derived from golden chips at different process corners, in order to detect the presence or absence of a Trojan. Next we describe each of the steps in detail.

Functional Decomposition: The first step of the proposed self-referencing approach is decomposition of a large design into functional blocks or regions. Sometimes, the circuit under test is designed with clearly-defined functional blocks which can be selectively activated by using control signals, like the 4-bit ALU circuit which we considered for our example in Section 2. Another type of circuit which is amenable to simple functional decomposition is a pipelined processor, where the different pipeline stages correspond to the different regions. However, there can be circuits which are available as a flattened gate-level netlist. For this we could use a hyper-graph based approach to identify partitions which have minimum cut-sets between them. This allows us to isolate the activity in one partition from causing activity in other regions. The region-based partitioning described in [7] can also be used for creating partitions in circuits which do not have well-defined functional blocks or for creating sub-blocks within a functional block. The decomposition should follow a set of properties to maximize the effectiveness of the approach:

1. The blocks should be reasonably large to cancel out the effect of random parameter variations, but small enough to minimize the background current. It should also be kept in mind that if the regions are too small, the number of regions can become unreasonably large for the test vector generation algorithm to handle.
2. The blocks should be functionally as independent of each other as possible so that the test generation process can increase the activity of one block (or few blocks) while minimizing the activity of all others.
3. The decomposition process can be performed hierarchically. For instance, a system-on-a-chip (SoC) can be divided into the constituent blocks which make up the system. But, for a large SoC, one of the blocks could itself be a processor. Hence, we need to further divide this structural block into functional sub-blocks.

Statistical test vector generation: In order to increase the Trojan detection sensitivity, proper test vector generation and application are necessary to reduce the background activity and amplify the activity inside the Trojan circuit. If we partition the circuit into several functional and structurally separate blocks, we can activate them one at a time and observe the switching current for that block with respect to the current values for other blocks. The test vector generation algorithm needs to take into account two factors:

1. Only one region must be activated at a time. If the inputs to different modules are mutually exclusive and the regions have minimal interconnection, it is easy to maximally activate one region while minimizing activity in other regions. If complex interconnections exist between the modules, the inputs need to be ranked in terms of their sensitivity towards activating different modules and the test generation needs to be aware of these sensitivity values.
2. When a particular region is being activated, the test vectors should try to activate possible Trojan trigger conditions and should be aimed at creating activity within most of the innumerable possible Trojans. This motivates us to consider a statistical test generation approach like the one described in [12] for maximizing Trojan trigger coverage. Note that, unlike functional testing approaches, the Trojan payload need not be affected during test time, and the observability of Trojan effect on the side-channel parameter is ensured by the region-based self-referencing approach described earlier.

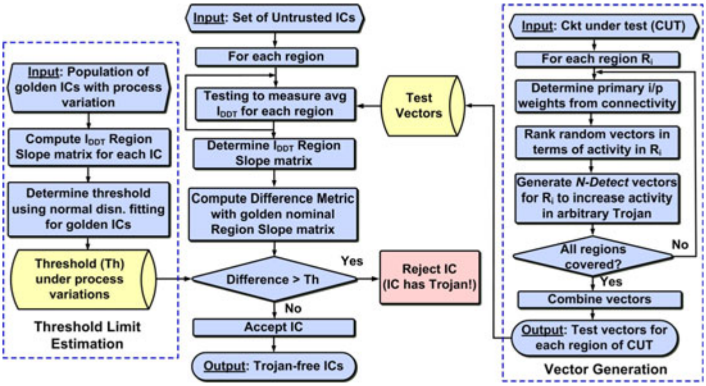
Input: Population of golden ICs with process variation
Compute I_DDT Region Slope matrix for each IC
Determine threshold using normal disn. fitting for golden ICs
Threshold (Th) under process variations
Threshold Limit Estimation
Input: Set of Untrusted ICs
For each region
Testing to measure avg I_DDT for each region
Determine I_DDT Region Slope matrix
Compute Difference Metric with golden nominal Region Slope matrix
Difference > Th
Yes
No
Reject IC (IC has Trojan!)
Accept IC
Output: Trojan-free ICs
Test Vectors
Input: Ckt under test (CUT)
For each region R_i
Determine primary i/p weights from connectivity
Rank random vectors in terms of activity in R_i
Generate N-Detect vectors for R_i to increase activity in arbitrary Trojan
All regions covered?
No
Yes
Combine vectors
Output: Test vectors for each region of CUT
Vector Generation

Fig. 4. The major steps of the proposed self-referencing methodology. The steps for test vector generation for increasing sensitivity and threshold limit estimation for calibrating process noise are also shown.

Fig. 4 shows a flow chart of the test vector generation algorithm on the right. For each region, we assign weights to the primary inputs in terms of their tendency to maximize activity in the region under consideration while minimizing activity in other regions. This step can also identify control signals which can direct the activity exclusively to particular regions. Next, we generate weighted random input vectors for activating the region under consideration and perform functional simulation using a graph-based approach, which lets us estimate the activity within each region for each pair of input vectors. We sort the vectors based on a metric C_{ij} which is higher for a vector pair which can maximally activate region R_i while minimizing activity in each of the other regions. Then, we prune the vector set to choose a reduced but highly efficient vector set generated by a statistical approach such as *MERO* [12]. In this approach (motivated by the *N-detect* test generation technique), within a region, we identify internal nodes with rare values, which can be candidate trigger signals for a Trojan. Then we identify the subset of vectors which can take the rare nodes within the region to their rare values at least N times, thus increasing the possibility of triggering the Trojans within the region. Once this process is completed for all the regions, we combine the vectors and generate a test suite which can be applied to each chip for measuring supply current corresponding to each of its regions.

For functional test of a multi-core processor, we can use specially designed small test programs which are likely to trigger and observe rare events in the system such as events on the memory control line or most significant bits of the datapath multiple times. In general a design is composed of several functional blocks and activity in several functional blocks can be turned off using input conditions. For example in a processor, activity in the floating point unit (FPU), branch logic or memory peripheral logic can be turned off by selecting an integer ALU operation. Many functional blocks are pipelined. In these cases, we will focus on one stage at a time and provide initialization to the pipeline such that the activities of all stages other than the one under test are minimized by ensuring that the corresponding stage inputs do not change. Next we describe how the self-referencing approach can be applied to compare the current values for different regions and identify the Trojan-infected region.

Side-Channel Analysis using Self-Referencing: In this step, we measure the current from different blocks which are selectively activated, while the rest of the circuit is kept inactive by appropriate test vector application. Then the average supply current consumed by the different blocks is compared for different chip instances to see whether the relations between the individual block currents are maintained. Any discrepancy in the "slope" of the current values between different blocks indicates the presence of Trojan. This approach can be hierarchically repeated for further increasing sensitivity by decomposing the suspect block into sub-blocks and checking the self-referencing relationships between the current consumed by each sub-block.

The flowchart for this step is shown in Fig. 4. Note that the best Trojan detection capability of region-based comparison will be realized if the circuit is partitioned into regions of similar size. The *Region Slope Matrix* is computed by taking the relative difference between the current values for each region. We estimate the effect of process variations on the "slopes" to determine a threshold for separating the golden chips from the Trojan-infested ones. This can be done by extensive simulations or measurements from several known-golden chips. For a design with n regions, the *Region Slope Matrix* is an $n \times n$ matrix, with entries that can be mathematically expressed as:

$$S_{ij} = \frac{I_i - I_j}{I_i} \; \forall i, j \in [1, n] \tag{2}$$

For each region, we get $2n - 1$ slope values, of which one of them is '0', since the diagonal elements S_{ii} will be zero.

The intra-die systematic variation is eliminated primarily because we use the current from an adjacent block, which is expected to suffer similar variations, to calibrate process noise of the block under test. The intra-die random variations can be eliminated by considering switching of large number of gates. In our simulations we find that even switching of 50 logic gates in a block can effectively cancel out random deviations in supply current.

Decision Making Process: In this step, we make a decision about the existence of Trojan in a chip. The variation in slope values for different regions for a chip from the golden nominal values are combined by taking the L^2 norm (sum of squares of difference of corresponding values) between the two Region Slope matrices. This difference metric for any chip 'k' is defined as

$$D(k) = \sum_{i=1}^{N} \sum_{j=1}^{N} (S_{ij}|_{Chip\ k} - S_{ij}|_{golden,nominal})^2. \tag{3}$$

The limiting "threshold" value for golden chips can be computed by taking the difference $D(golden, process\ variations)$ as defined by

$$Threshold = \sum_{i=1}^{N}\sum_{j=1}^{N}(S_{ij}|_{golden,process\ variation} - S_{ij}|_{golden,nominal})^2. \quad (4)$$

Any variation beyond the threshold is attributed to the presence of a Trojan. The steps for computing the golden threshold limits are illustrated on the left side of Fig. 4. Since unlike conventional testing, a go/no-go decision is difficult to achieve, we come up with a measure of confidence about the trustworthiness of each region in a chip using an appropriate metric. We compare the average supply current consumed by the different blocks for different chip instances to see whether the expected correlation between the individual block currents is maintained. The Trojan detection sensitivity of the self-referencing approach can be defined as

$$Sensitivity = \frac{D(tampered, nominal)}{Threshold} \quad (5)$$

Since, the slope values are less affected by process variations compared to the current values alone, we expect to get better sensitivity compared to eqn. (1). Note that since we perform region-based comparison, we can localize a Trojan and repeat the analysis within a block to further isolate the Trojan. This approach can be hierarchically repeated to increase the detection sensitivity by decomposing a suspect block further into sub-blocks and applying the self-referencing approach for those smaller blocks. We can also see that the region-based self-referencing approach is scalable with respect to design size and Trojan size. For the same Trojan size, if the design size is increased two-fold, we can achieve same sensitivity by dividing the circuit into twice as many regions. Similarly we can divide the circuit into smaller regions to increase sensitivity towards detection of smaller Trojan circuits.

4 Results

4.1 Simulation Results

We used two test cases to validate the proposed Trojan detection approach: 1) a 32-bit integer Arithmetic Logic Unit (ALU), and 2) a Finite Impulse Response (FIR) digital filter. The size of the ALU circuit can be scaled by changing the word size parameter. We considered 4 structurally different blocks - adder (*add*), subtracter (*sub*), multiplier (*mul*) and shifter (*shift*) which can be selectively activated by the *opcode* input bits. However, the FIR filter had a flattened netlist and was manually partitioned into four regions with the minimum interconnections, and the test vector generation tool (written in MATLAB) was used to generate test vectors to selectively activate each block. We inserted a small (<0.01% of total area) Trojan in the subtracter of the ALU and the 4^{th}

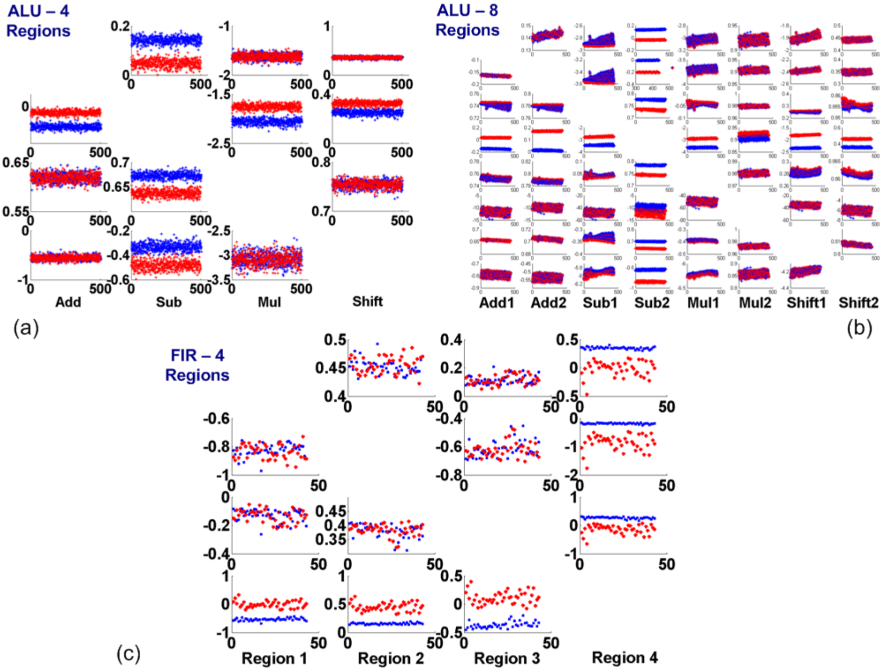

Fig. 5. Self-referencing methodology for detecting Trojan in the 32-bit ALU and FIR circuits. Blue and red lines (or points) denote golden and Trojan chips, respectively.

region of the FIR filter. Both designs were synthesized using Synopsys *Design Compiler* and mapped to a LEDA standard cell library. Circuit simulations were carried out for the 70nm *Predictive Technology Model* (PTM) [15] using Synopsys *HSPICE*. To estimate the effect of process variations, we used Monte Carlo simulations for a maximum of ±20% variation in the nominal V_T value, inter-die variations with $\sigma = 10\%$ and random intra-die variations with $\sigma = 6\%$. We simulated the circuits and separately measured the supply current for different regions for 500 golden chips and 500 infected chips.

The simulated *Region Slope Matrix* values are plotted in Fig. 5(a). The Trojan-infected chip instances can be easily distinguished from the golden ones, even in the presence of process noise. The row and column corresponding to the subtracter (2^{nd} region) show visibly different values for the golden (blue) and Trojan (red) values. Next, we performed simulations with multiple vector pairs activating the same module to show that the Trojan in the subtracter is only selectively activated on the application of one of the two vector pairs activating the subtracter module. The *Region Slope Matrix* for this case is shown in Fig. 5(b). This matrix contains 8 regions since each of the four structurally separate regions of the ALU are further divided into two sub-blocks, corresponding

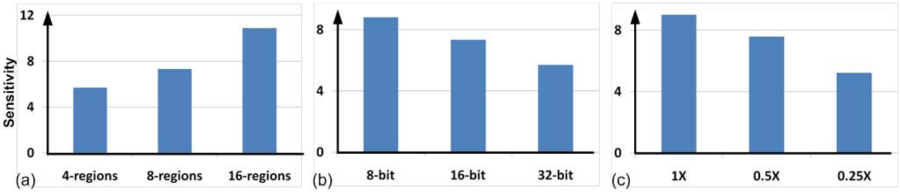

Fig. 6. Sensitivity analysis with (a) different number of regions, (b) different circuit sizes, and (c) different Trojan sizes

Table 1. Probability of Detection and probability of False Alarm (False Positives)

Circuit Name	TN(%)	FP(%)	FN(%)	TP(%)
32-bit ALU	99.10	0.90	5.90	94.10
FIR	97.72	2.28	6.60	93.40

to the two different vector pairs which share the same opcode values. It can be readily observed that increasing the number of regions increases the sensitivity of Trojan detection.

Fig. 5(c) shows the simulation results for the FIR design. The test vectors are chosen by the MATLAB tool and used to dominantly activate different regions of the design. The *Region Slope Matrix* is computed for 50 golden chips and 50 Trojan-infected chips and we can successfully detect the Trojan-infected region (region 4). Fig. 6 shows the variation in sensitivity of the self-referencing approach by varying different parameters of the ALU. For a 16-bit ALU, we see that increasing the number of regions helps increase the sensitivity in Fig. 6(a). In Fig. 6(b), we plot the sensitivity of the approach for increasing circuit sizes. Finally in Fig. 6(c), we show that increasing the number of regions also helps to keep the sensitivity nearly constant as we scale down the Trojan size. The percentage of true positives, true negatives, false positives and false negatives as obtained from the Monte Carlo simulations are presented in Table 1. We used a process point with 20% V_T variation to compute the threshold. For smaller circuits and larger Trojans the sensitivity is higher and hence, the accuracy of classification is also better.

4.2 Experimental Results

We used a custom test board with socketed Xilinx Virtex-II XC2V500 FPGAs to measure current from eight individual supply pins as well as the total current, using 0.5Ω precision current sense resistors to sense the I_{DDT} and an Agilent mixed-signal oscilloscope (100MHz, 2 Gsa/sec) to record the data. The test circuit was a 32-bit DLX processor with a 5-stage pipeline which contains the previously-described 32-bit ALU as part of its execution unit, occupying over 80% of the FPGA slices. The Trojan circuit was a 16-bit serial-in parallel-out

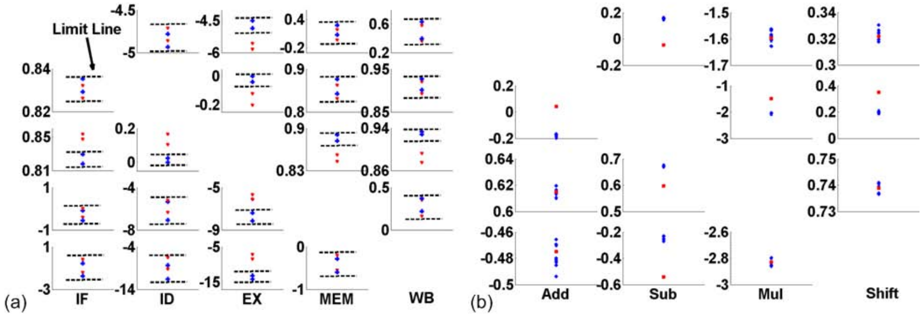

Fig. 7. Experimental results for 8 golden and 2 tampered FPGA chips. *Region slope matrix* for (a) 32-bit DLX processor; (b) 32-bit ALU. The limit lines are obtained by analyzing the 8 golden chips. The red points denote the values for the Trojan-containing test chips while the blue points denote the values for the golden chips.

shift register (sequential Trojan) occupying 0.08% of total area. We performed experiments with 10 FPGA chips from the same lot. We insert a Trojan in two of the ten chips inside the subtracter sub-region of the ALU. The *Region Slope Matrix* is constructed using the measured current values for the five pipeline stages of the DLX processor in the 10 FPGA chips. We use the 8 golden chips to determine the threshold limit and use our self-referencing approach to test 4 test chips (2 golden and 2 Trojan). As can be clearly seen from Fig. 7, the Trojan containing chips are easily identified as well as the region which contains the Trojan in both cases. Next, we repeat the procedure using test vectors which only activate the four sub-regions inside the 32-bit ALU and identify that the Trojan is located within the subtracter.

5 Conclusion

We have presented a side-channel hardware Trojan detection approach that exploits the intrinsic relationship between active-mode current among the different regions of a chip to achieve high signal-to-noise ratio in presence of process variations. We have shown that the self-referencing approach coupled with efficient vector generation provides scalability in terms of increasing process variations (thus being amenable to future scaled technologies) and increasing design size. As a by-product, such an approach also helps to localize the Trojan, which can be helpful for diagnosis. Simulation results for different circuits are supported by the experimental validation for a 32-bit DLX processor core. The approach can be easily extended to multi-core SoC, where the cores can be hierarchically partitioned into multiple regions or functional units. Another possible application involves detecting instances of re-marked chips in a lot of manufactured ICs, which pass functional testing but can cause in-field failure.

References

1. DARPA: TRUST in Integrated Circuits, TIC (2007), `http://www.darpa.mil/MTO/solicitations/baa07-24`
2. Adee, S.: The hunt for the kill switch. IEEE Spectrum 45(5), 34–39 (2008)
3. King, S., et al: Designing and implementing malicious hardware. In: LEET (2008)
4. Wolff, F., et al.: Towards Trojan-free trusted ICs: Problem analysis and detection scheme. In: DATE, pp. 1362–1365 (2008)
5. Chakraborty, R.S., Narasimhan, S., Bhunia, S.: Hardware Trojan: threats and emerging solutions. In: HLDVT (2009)
6. Rad, R., Plusquellic, J., Tehranipoor, M.: A sensitivity analysis of power signal methods for detecting hardware Trojans under real process and environmental conditions. IEEE Tran. VLSI (2010)
7. Banga, M., Hsiao, M.: A region based approach for the identification of hardware Trojans. In: HOST, pp. 40–47 (2008)
8. Adamov, A., Saprykin, A., Melnik, D., Lukashenko, O.: The problem of hardware Trojans detection in system-on-chip. In: CADSM, pp. 178–179 (2009)
9. Agrawal, D., Baktir, S., Karakoyunlu, D., Rohatgi, P., Sunar, B.: Trojan detection using IC fingerprinting. In: Symposium on Security and Privacy, pp. 296–310 (2007)
10. Rad, R., Wang, X., Tehranipoor, M., Plusquellic, J.: Taxonomy of Trojans and methods of detection for IC trust. In: ICCAD (2008)
11. Jin, Y., Makris, Y.: Hardware Trojan detection using path delay fingerprint. In: HOST (2008)
12. Chakraborty, R.S., Wolff, F., Paul, S., Papachristou, C., Bhunia, S.: MERO: A statistical approach for Hardware Trojan detection. In: CHES (2009)
13. Borkar, S., et al.: Parameter variations and impact on circuits and microarchitecture. In: DAC, pp. 338–342 (2003)
14. Papoulis, A., Pillai, S.U.: Probability, Random Variables and Stochastic Processes, 4th edn. McGraw-Hill, New York (2002)
15. Predictive Technology Model, `http://www.eas.asu.edu/~ptm/`

Appendix

Analysis of the Effect of Process Variations. In order to increase the hardware Trojan detection sensitivity for a large design with ultra-small Trojan, we need to amplify the Trojan effect while nullifying the impact of process variations in the side-channel parameter. There are two types of process variations [13] – *inter-die* variations and *intra-die* variations, with the latter having a systematic component and a random component. Inter-die variations are the parameter variations from one die to another on a wafer and can be modeled by a variation in the transistor threshold voltage (V_T) for the entire design. Intra-die variations are the variations within the same die which can cause different parametric variations than that predicted by inter-die variations. They have a random component which causes random variation in V_T of each transistor about the V_T of the die. There can also be a systematic component to these variations since there are spatial correlations among the V_T variations of the transistors. Fig. 8 shows the effect of the different components of process variation on the V_T of devices in

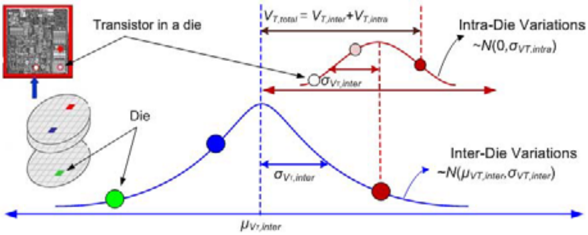

Fig. 8. The effect of process variation on device threshold voltage in an IC

an IC, where each of the "inter-die" and "intra-die" components are modeled as *normal distribution* with certain mean (μ) and standard deviation (σ).

Consider an IC that has been partitioned into N different regions, such that each region can be preferentially activated while the activity of the other partitions are minimized. Consider that the region R_i has been preferentially activated, and consider a gate $g \in R_i$. Then, the switching current of g is approximately given by $I_g = k(V_{DD} - V_{Tg})^2$, where k is a constant depending on the process and the nature of the gate, V_{DD} is the supply voltage and V_{Tg} is the threshold voltage of the i-th gate. Now, V_{Tg} can be expressed as $V_{Tg} = V_T + \Delta V_{Ti} + \Delta v_{Tg1} + \Delta v_{Tg2}$, Here, ΔV_{Ti} represents the effect of the "systematic intra-die" component of variation, and has the same value for all gates in the region R_i; Δv_{Tg1} represents the effect of the "inter-die" component of process variation, and has the same value for all gates in the IC, and Δv_{Tg2} is the effect of the "random intra-die" component of process variation, and has random values for different gates of the IC. Hence,

$$\begin{aligned} I_g &= k\left[V_{DD} - (V_T + \Delta V_{Ti} + \Delta v_{Tg1} + \Delta v_{Tg2})\right]^2 \\ &= k\left[(V_{ov} - \Delta v_{Tg1})^2 + (\Delta V_{Ti} + \Delta v_{Tg2})^2 - 2\left(V_{ov} - \Delta v_{Tg1}\right)\left(\Delta V_{Ti} + \Delta v_{Tg2}\right)\right] \end{aligned} \tag{6}$$

where $V_{ov} = V_{DD} - V_T$ is the *gate overdrive*. Ignoring all second order terms involving both random and systematic shifts of the threshold voltage, the above equation can be approximated by:

$$I_g \approx \underbrace{k\left[V_{ov}^2 - 2V_{ov}(\Delta v_{Tg1} + \Delta V_{Ti})\right]}_{\text{constant for each gate } g \in R_i} - \underbrace{2V_{ov}\Delta v_{Tg2}}_{\text{random for each gate } g \in R_i} \tag{7}$$

Summing the currents for all the switching gates of the region R_i, the total switching current for region R_i is:

$$I_i = \sum_{g \in R_i} I_g = kn_i\left[V_{ov}^2 - 2V_{ov}(\Delta v_{Tg1} + \Delta V_{Ti})\right] - 2V_{ov}\sum_{g \in R_i} \Delta v_{Tg2} \tag{8}$$

where n_i is the number of switching gates in region R_i. Now, the term $\sum\limits_{g \in R_i} \Delta v_{Tg2}$ represents the sum of n_i (normally distributed) random variables, each with mean $\mu = 0$ and standard deviation σ_T (let). Hence, by the *Central Limit Theorem* [14], the term $\sum\limits_{g \in R_i} \varDelta v_{Tg2}$ is approximately normally distributed with mean $\mu = 0$ and a reduced standard deviation $\frac{\sigma_T}{\sqrt{n_i}}$. Hence, for reasonably large value of n_i, this term is approximately equal to zero, and the expression for I_i can be approximated by:

$$I_i \approx \sum_{g \in R_i} I_g = kn_i \left[V_{ov}^2 - 2V_{ov}(\varDelta v_{Tg1} + \varDelta V_{Ti}) \right] \tag{9}$$

Similarly, for a region R_j. the switching current is given by:

$$I_j \approx \sum_{g \in R_j} I_g = kn_j \left[V_{ov}^2 - 2V_{ov}(\varDelta v_{Tg1} + \varDelta V_{Tj}) \right] \tag{10}$$

Hence, the difference between the currents of regions R_i and R_j can be expressed as:

$$\begin{aligned} I_i - I_j|_{observed} &= k \left[V_{ov}^2 - 2V_{ov}\varDelta v_{Tg1} \right] (n_i - n_j) - 2kV_{ov}(n_i \varDelta V_{Ti} - n_j \varDelta V_{Tj}) \\ &= c_1(n_i - n_j) + \underbrace{c_2(n_i \varDelta V_{Ti} - n_j \varDelta V_{Tj})}_{\text{due to } \textit{systematic intra-die} \text{ variation}} \end{aligned} \tag{11}$$

where c_1, c_2 are constants. If the contribution due to the intra-die systematic component is negligible, the above expression can be re-written as:

$$I_i - I_j|_{observed} \approx c_1(n_i - n_j) \qquad \text{and} \qquad I_i|_{observed} \approx c_1 n_i \tag{12}$$

Hence, the mutual *Region Slope* metric for regions R_i and R_j is

$$S_{ij,observed} = \frac{I_i - I_j}{I_i} = \frac{n_i - n_j}{n_i} \tag{13}$$

In the *nominal* case, in the absence of any process variation effects, $\varDelta V_{Ti} = \varDelta V_{Tj} = \varDelta v_{Tg1} = \varDelta v_{Tg2} = 0$; hence , $I_i - I_j|_{golden} = c_3(n_i - n_j)$, $I_i = c_3 n_i$ and

$$\boxed{S_{ij,golden} = \frac{n_i - n_j}{n_i} = S_{ij,observed}} \tag{14}$$

Similarly, it can be shown that $S_{ji,golden} = S_{ji,observed}$. This shows that under negligible *systematic intra-die* variations, the ratio of the difference in the switching currents of two regions and the current of each region should remain approximately unchanged. This equality fails to be satisfied in case one of the regions is modified by the insertion of a Trojan, because then the switching current of the gates constituting the Trojan circuit disturbs the balance. This observation is the main motivation behind using the Region Slope values for reducing the process noise. For a circuit with N regions, if we compute the Region Slope values for all pairs of regions, we obtain an $N \times N$ matrix, with zero diagonal elements. It is observed that systematic variations still cause some variations in the Region Slope values, but the effect of process variation has been reduced greatly compared to the variations in individual current values, thus giving us improved sensitivity for Trojan detection.