VITAMIN: Voltage Inversion Technique to Ascertain Malicious Insertions in ICs

Mainak Banga and Michael S. Hsiao
Bradley Department of Electrical and Computer Engineering
Virginia Tech, Blacksburg, Virginia - 24061
Email: {banga, mhsiao}@vt.edu

***Abstract*—We propose an *inverted voltage* scheme for exciting and pronouncing the behavior of any undesirable logic that may be inserted in the IC manufactured abroad. The *inverted voltage* scheme is coupled with a *sustained vector simulation* technique to further enhance the behavioral difference between the genuine and targeted test IC. Experimental results on a variety of ISCAS'89 benchmarks show that we are able to significantly magnify the difference between the genuine and tampered designs. For most of the smaller benchmarks, our inverted voltage method is able to detect the effect of the tamper directly at the primary outputs of the IC, especially when existing techniques fail to make any observable distinction. For the larger circuits, our technique is able to magnify the power consumption of the tampered circuit by several times.**

I. Introduction

The pressures in manufacturing cost and time to market have exacerbated the outsourcing of fabrication process to overseas fabrication centers owned and managed by a third party. Since these fabrication units are not under direct surveillance of design companies, genuineness of the final product relies on the intent of the fabricator. A fabricator with a malicious intent can introduce subtle alterations in the design by adding an extraneous logic or modifying an existing logic [1]. Such undesirable implantation(s) are called as *Hardware Trojan(s)* or simply *Trojans*. So, finished chips imported from a third party manufacturer must be tested for possible tampers before they can be deployed for mission critical applications like missile systems, medical equipments, space exploration etc.

Conventional design-for-testability (DFT) like scan-based testing and *Built-In-Self-Test* (BIST) [2], have been used to enhance the testability of an IC thereby ensuring that a defective part is sieved out before the final product is shipped to customer. However, since these test patterns are intended to detect defects based on existing fault models, they cannot necessarily excite/detect intentional Trojans because of their unknown, possibly stealthy triggering condition.

Most of the contemporary *Trojan* detection methods monitor one or more side-channel signal(s) for assessing behavior of the circuits under test (CUTs). Extraction of internal information like security keys using side-channel signal analysis [3], [4], can lead to misuse of an IC [5]. For protecting the rights of IP designer, features are embedded in the IC to shield it from piracy or side-channel signal attacks. Security schemes based on *Physically Unclonable Functions* (PUF) [6], *Digital Watermarking* etc. have been proposed towards this end. Nevertheless, if any third party intends to degrade the performance of underlying design tactfully, he/she can implant *Trojan(s)* even in the presence of these security features.

In the recent past, a number of methods have been proposed to detect the presence of *Trojan(s)* inside a design. In [7] the authors simulate a set of random vectors on the genuine and *Trojan* affected ICs and observe the power profile in both cases. Circuits affected with *Trojan(s)* consume more power but this difference is not truly discernible unless the circuit activity is kept really low. In [8] the authors have used a current integration technique to capture the prolonged effect of *Trojan(s)* on total charge consumption of the device. In [9], partition based schemes are employed to target and activate circuit portions individually. In [10], a sustained vector set is used to ensure that the circuit activity is generated by state transitions only. Results show that these methods are effective for circuits whose switching activity can be kept low.

In this work we propose an inverted voltage scheme that aims to activate the *Trojan(s)* with a much higher triggering frequency. Once triggered/activated, the effect of a *Trojan* becomes more prominent. To strengthen its effectiveness, we integrate this approach with the sustained vector technique introduced in [10] to magnify the power profile difference. To make Trojans hard to detect, our experimental *Trojan* consists of a single gate only. For the smaller benchmarks we were able to detect the effect of *Trojan* directly on the primary outputs using our approach. For the rest of the benchmarks, the power profile behavior shows significantly increased variations between the genuine and the *Trojan*-affected designs, the differences were enhanced by many times, sometimes in orders of magnitude.

II. Preliminaries

A. Effects of Voltage Inversion on CMOS

Complimentary Metal Oxide Semiconductor (CMOS) is the most widely used logic family in today's semiconductor industry [11]. Any CMOS gate consists of two complimentary networks - a *pull-up* network containing the PMOS gates and a *pull-down* network containing the NMOS gates. Since PMOS gates can pass a clean "1" (they don't degrade the voltage with noise), they are used to construct the *pull-up* network. On the other hand, NMOS gates can pass a clean "0" making them ideal for constructing the *pull-down* network. If a PMOS gate is used to pass "0", the voltage at the output terminal is not exactly "0". Instead it is $V_{TH} > 0$ which is the *threshold voltage*. Similarly, if an NMOS gate is used to pass a "1"

This work is supported in part by NSF grant 0840936

978-1-4244-4804-3/09/$25.00 © 2009 IEEE

the voltage at the output terminal is $V_{DD} - V_{TH} < V_{DD}$. This is true for a chain of NMOS or PMOS gates as well. Across a chain of PMOS gates in which the drain of one gate is connected to the source of the next gate, there is a rise of V_{TH} at the final output while passing a "0". Likewise for a chain of NMOS gates there is a drop of V_{TH} at the final output while passing a "1". If we treat V_{TH} to be small enough to be approximated as "0" and $V_{DD} - V_{TH}$ to be high enough to be approximated as "1", the new gates resulting from the application of inverted voltages show that INVERTER behaves like BUFFER, NAND gate behaves like AND gate and NOR gate behaves like OR gate, i.e., their logical functionalities are inverted.

B. Power Profile

Power consumption has been shown as an effective side-channel signal of interest. Circuit power can be considered as the sum of static power, dynamic power and leakage power. Of these, the variable component is the dynamic power. Dynamic power is related to the frequency and gate capacitance as per the following equation:

$$\mathbf{P = CV^2f} \tag{1}$$

where P denotes *dynamic power*, C denotes switching capacitance, V denotes supply voltage and f denotes frequency of operation of the circuit. As supply voltage V and operating frequency f remain constant for an IC, the parameter of interest is switching capacitance C depends on number of gates toggling in a circuit for any given input vector pair.

We model a circuit as a levelized structure. We start by assigning the inputs and flip-flops a level of 0. All other gates in the circuit are assigned a level 1 greater than the maximum level of set of its input gates. This way, every gate in the circuit is assigned either an *even level* or and *odd level*. In our methodology, we propose that the gates in the alternate levels are supplied V_{DD} and GND connections using separate voltage supply networks. Of course, the supply voltage of the flip-flops remain unchanged. To perform normal operation, supply pins of both levels of gates are connected to V_{DD} and ground pins to the GND. When one of the levels is selected to be operated under opposite voltage supplies, the polarities of the corresponding pins connecting to these rails can be changed.

III. Our Approach

Our approach is based on two principles - (1) *Gate Logic Inversion* creates a frequent triggering scenario for the *Trojan* gate and (2) *Sustained Vector Simulation* aims towards minimizing the overall circuit activity to magnify the extra toggles created by the *Trojan*.

For any gate g, if a logic value $L(L\epsilon\ \{0,1\})$ appears more frequently at its output under a vector set T than the other logic value $\overline{L}$, we call L as the *majority value* and $\overline{L}$ as the *minority value* for g under T. For instance, under a random test set, logic 0 is generally the *majority value* for an AND gate and *minority value* for an OR gate, while logic 1 is the

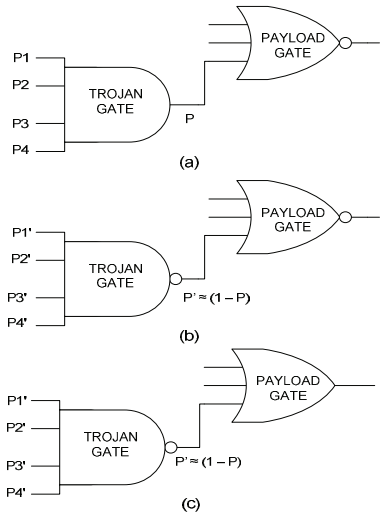

Fig. 1. (a) *Trojan* logic under normal voltage supply (b) *Trojan* logic where only *Trojan* gate is affected by inverted voltage supply (c) *Trojan* logic where both *Trojan* and *Payload* gate is affected by inverted voltage supply

minority value for an AND gate and *majority value* for an OR gate. Triggering of a *Trojan* generally requires the attainment of *minority value* at its output, otherwise it will be frequently triggered and easily detectable. Furthermore, the probability of attainment of the *minority value* by a *Trojan* depends on setting the necessary values on its inputs. Under normal operating conditions, a *Trojan* is difficult to trigger because the input combination that triggers it is difficult to attain.

Let us consider a 4-input *Trojan AND* gate shown in Fig. 1 (a). It feeds its output to an existing NOR gate in the genuine circuit which we call as the *Payload* gate. Let $P1$, $P2$, $P3$ and $P4$ be the probabilities that the respective inputs of the *Trojan* gate can be set to non-controlling value 1. So the probability that the output is set to *minority value* 1 can be approximated as $P1 \times P2 \times P3 \times P4$, assuming signal independence of the four inputs of the AND gate. For a general *Trojan* gate with n inputs, if $P[i]_{NC}$ be the probability that an input i to the *Trojan* attains a non-controlling value, then the probability that the *Trojan* is triggered for a given combination of inputs can be approximated by:

$$P(trigger) = \prod_{i=0}^{i=n} P[i]_{NC} \tag{2}$$

Now let us consider the case when the supply voltage to the *Trojan AND* gate is inverted while the supply of the *Payload* gate remains same, shown in part (b) of Fig. 1. From the preceding discussion we know that when we invert the supply voltage of a gate, it behaves as its complement. Thus the AND gate is converted into a NAND gate detonating its payload to the NOR gate. The controlling value for the NOR gate is 1 which is the *majority value* for NAND gate. Thus under the inverted voltage conditions, the triggering scenario occurs more frequently (due to the *Trojan* gate) than in the normal operation. The situation where both the *Trojan* and *Payload*

gate gets inverted simultaneously is shown in Fig. 1 (c). In this case, the NOR gate is converted to an OR gate at the output of the *Trojan* NAND gate. Since 1 is still the controlling value for OR gate, the *Trojan* would continue to provide the triggering value. Although the probability of a *minority value* assignment at the output of the *Trojan* in the *inverted voltage* mode will not be exactly $(1 - P)$ (owing to change in the input probabilities because of voltage inversion), the *minority value* and *majority value* gets swapped causing more triggers.

Inverting the voltage for the entire circuit in one go causes the voltage drop to build up across successive levels, i.e., the degraded gate potential of one stage of CMOS gates degrades the gate potential on the next stage. This way, after a few stages, the signal voltage may degrade beyond the capability of further propagating any activity, stalling the circuit. So we apply *inverted voltage* to alternate levels in the circuit. Feeding gates in alternate levels with different supply voltages ensure that the drop in voltage at the output of a gate operating in *inverted voltage* domain is compensated by gate(s) in its immediate fanout cone (which operate(s) in the normal voltage domain and hence swings their output rail-to-rail). In our experiments, we first simulate the circuit without any inversion (phase I). Then, we let the odd-level gates be inverted, apply test vectors (phase II), followed by making the odd-level gates normal and even-level gates inverted (phase III). Note that in both cases the flip-flops are never inverted. The *Trojan* gate, if connected to the supply of the odd level or the even level will be inverted in at least one of the phases of testing (phase II or phase III). Since the output of the *Trojan* gate attains *majority value* for most of the time whenever it is in *inverted voltage* mode, it will inject a controlling value into the *Payload gate* that propagates down its fanout cone to create more extraneous activities. This is reflected in the power profile of the *Trojan* affected circuit because in genuine circuit such toggles will not occur. We reinforce the voltage inversion technique with a *sustained vector simulation* to further enhance the behavioral discrepancies. For a detailed description of *sustained vector technique* interested readers should refer to [10].

When a gate is in the *inverted voltage* domain, the output voltage does not swing rail to rail because of the *threshold voltage* drop. So the power consumed for a transition in *inverted voltage* domain is somewhat smaller than that in a normal voltage domain. For a gate in the normal voltage domain we count a transition as 1. For a gate in the inverted voltage domain a transition is scaled down by a factor to compensate for the reduced voltage swing. The voltage scaling factor SF is determined as per Equation 3.

$$SF = \frac{V_{DD(INV)} - V_{GND(INV)}}{V_{DD} - V_{GND}} \tag{3}$$

where $V_{DD(INV)}$ and $V_{GND(INV)}$ correspond to the maximum and minimum voltage for a gate in the *inverted voltage* mode and V_{DD} and V_{GND} denote the maximum and minimum voltage for a gate in the normal voltage mode respectively. Assuming a $1.8V$ supply as V_{DD}, $0V$ for GND and $0.2V$ for the *threshold voltage* V_{TH} and substituting $V_{DD(INV)} = V_{DD} - V_{TH}$ and $V_{GND(INV)} = V_{TH}$ in Equation 3, we get $SF \simeq 0.75$. Thus in our experiment we count a toggle at the output of a gate in the *inverted voltage* domain as 0.75.

Threshold voltage(V_{TH}) is a device characteristics. Its value depends on the gate oxide thickness. In [12] the authors have shown the variation of threshold voltage on the channel length as well as on the drain voltage. It is possible to control the device processing parameters so that the *threshold voltage* is only a small fraction of the supply voltage. This ensures that the drop occurring at the output of a *pull-up* or *pull-down* network is not high enough to degrade the signal strength beyond recognition at the input of the next stage.

IV. Experimental Results

In all our experiments, we have used a single 4-input gate *Trojan*. The single gate can be either an AND gate or an OR gate, inserted into a variety of ISCAS89 sequential benchmark circuits. Without voltage inversion, these Trojans were extremely hard to detect. The experiments were carried out on a 3GHz Intel dual-core quad processor machine with 2GB RAM. The entire code for test generation is written in C++. The typical test generation time varied from a few seconds for the small circuits to a few minutes for the larger benchmarks. All *Trojans* used in our experiments were tested to be non discernible at the output using the set of 10000 random vector sequence.

Fig. 2 to Fig. 5 show the relative increase in percentage activity of *Trojan* affected circuit over the genuine circuit for an AND/OR *Trojan* when the CUT is simulated with random vectors and sustained random vectors respectively. In both of these cases, magnification obtained by using random vectors on the *inverted voltage* set up are significantly higher when compared to the *normal voltage* mode. In almost all cases, the random vectors failed to create a difference in excess of 5% which is typically considered as the process variation threshold [7]. With sustained vectors, we can achieve substantial improvement in creating a measurable power difference when applied on top of the proposed *inverted voltage* methodology. In most cases, as shown in the Fig. 4 and Fig. 5, the difference is magnified by many times, sometimes by several orders of magnitude. The cases in which the *Trojans* were detectable at the output, the corresponding bar is marked by a **D**.

The exposure of the Trojan at a primary output usually happens for the smaller to medium sized benchmarks because of shorter propagation paths of triggered *Trojan* output signal to the circuit primary output. In larger benchmarks, triggered values may be blocked from reaching an output by a controlling value on some other gate in its propagation path. When the primary outputs can differentiate the genuine and Trojan circuits, no power analysis is needed. Yet for the circuits where we could detect the *Trojans* directly at the primary output, we still report the power difference values for completeness. Among all twenty-two Trojan circuits (11 AND + 11 OR-type Trojans), voltage inversion with random vectors was able to detect ten of them directly at the primary outputs. Note that without voltage inversion, no Trojan was

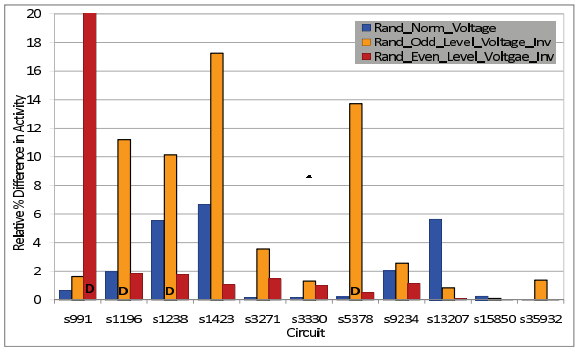

Fig. 2. % diff. in activity for AND Trojans with random vectors

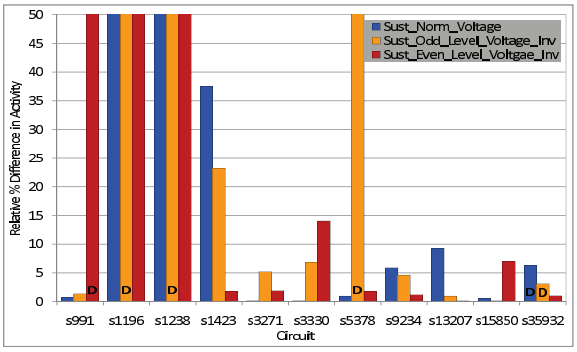

Fig. 4. % diff. in activity for AND Trojans with sustained random vectors

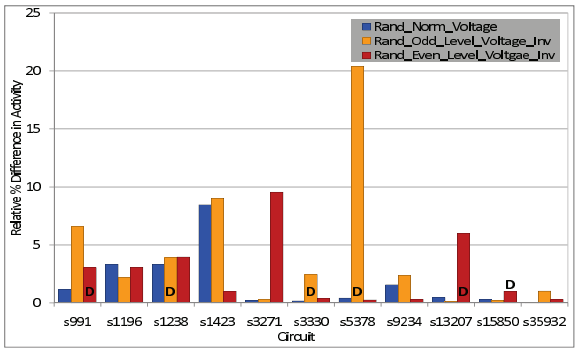

Fig. 3. % diff. in activity for OR Trojans with random vectors

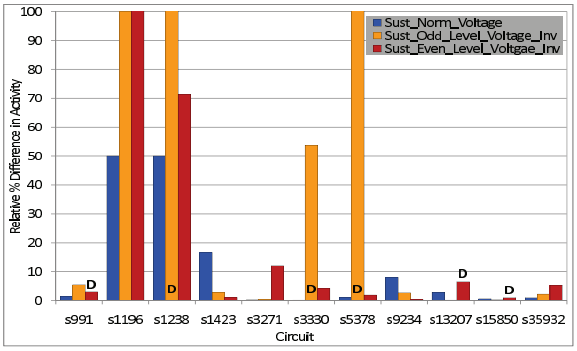

Fig. 5. % diff. in activity for OR Trojans with sustained random vectors

detectable. When sustained random vectors were used, we were able to detect eleven Trojan circuits directly at the output. In addition, with differential power analysis,we were able to detect all the remaining circuits as well. The results show that the power differential is indeed a robust measure to detect malicious activity irrespective of whether the output already differentiated the Trojan circuit from the genuine circuit!

For circuits such as $s3271$, $s3330$ and $s35932$, which are inherently very high toggling, previous methods have shown to be ineffective. As clear from the results, our technique proves to be very effective is creating an observable difference in these circuits as well. For $s9234$ with OR-type Trojan and for $s13207$ with AND-type Trojan, they cannot be detected at the output. However, the sustained random technique with normal voltage mode provides highest activity magnification. This is because changing the voltage supplies changes the logic and it is possible that such an alteration renders the circuit less sensitive to activities from the Trojan.

V. Conclusion and Future Works

We have presented a new *inverted voltage* technique to better activate and detect the malicious insertions in third party ICs. This setup is also coupled by a *sustained vector set* ensuring minimum genuine activity inside the design so that extraneous activity created by *Trojan* gets exaggerated. Experimental results on ISCAS'89 benchmarks prove that the approach is very effective in detecting the presence of very small *Trojans* inside the designs. Future work in this direction will be an actual estimation of the overhead involved in laying out the supply rails in order to meet the requirements of providing controlled voltages to the even and odd level gates.

References

[1] F. Wolff, C. Papachristou, S. Bhunia and R. S. Chakraborty, *Towards Trojan-Free Trusted ICs: Problem Analysis and Detection Scheme*, DATE, 2008, pp. 1362-1365.

[2] G. Hetherington, T. Fryars, N. Tamarapalli, M. Kassab, A. Hassan and J. Rajski; *Logic BIST for large industrial designs: real issues and case studies*, ITC, 1999, pp. 358-367.

[3] D. Agarwal, B. Archambeault, J. R. Rao and P. Rohatgi, *The EM side-channel(s)*, Int. Workshop CHES, 2002, Vol. 2523, pp. 29-45.

[4] P. C. Kocher, *Timing attacks on implantations of diffiehellman, rsa, dss and other systems In Neal Kobitz*, Proc. of CRYPTO, Vol. 1109 of Lecture Notes in Computer Science, pp. 104-113.

[5] I. Verbauwhede, K. Tiri, D. Hwang, and P. Schaumont, *Circuits and design techniques for secure ICs resistant to side-channel attack*, Int. Conf. on IC Design and Technology, 2006, pp. 1-4.

[6] J. Guajardo, S. S. Kumar, G. J. Schrijen and P. Tuyls, *Physical Unclonable Functions and Public-Key Crypto for FPGA IP Protection*, Int. Conf. on Field Prog. Logic and App., Aug. 2007, pp. 189-195.

[7] D. Agarwal et. al.; *Trojan Detection using IC Fingerprinting*, Proc. of the Symposium on Security and Privacy, 2007, pp. 296-310.

[8] R. Rad, J. Plusquellic and M. Tehranipoor, *Sensitivity analysis to hardware Trojans using power supply transient signals*, HOST, 2008, pp. 3-7.

[9] M. Banga and M. Hsiao; *A Region Based Approach for the Detection of Hardware Trojans*, HOST, 2008, pp. 43-50.

[10] M. Banga and M. Hsiao; *A Novel Sustained Vector Technique for the Detection of Hardware Trojans*, VLSI Design, 2009, pp. 327-332.

[11] Neil H. E. Weste and D. Harris, *CMOS VLSI Design: A circuits and Systems Perspective*, IIIed, Pearson Education (Addison Wesley), 2005.

[12] S. Narendra, V. De, S. Borkar, D. Antoniadis, A. Chandrakasan, *Full-chip sub-threshold leakage power prediction model for sub-0.18nm CMOS*, International Symposium on Low Power Electronics and Design, 2002, pp. 19 - 23.