Reconciling the IC Test and Security Dichotomy

O. Sinanoglu NYU Abu Dhabi
N. Karimi, J. Rajendran, R. Karri Polytechnic Institute of NYU
Y. Jin University of Central Florida
K. Huang, Y. Makris University of Texas at Dallas

***Abstract*—Many of the design companies cannot afford owning and acquiring expensive foundries and hence, go fabless and outsource their design fabrication to foundries that are potentially untrustwrothy. This globalization of Integrated Circuit (IC) design flow has introduced security vulnerabilities. If a design is fabricated in a foundry that is outside the direct control of the (fabless) design house, reverse engineering, malicious circuit modification, and Intellectual Property (IP) piracy are possible. In this tutorial, we elaborate on these and similar hardware security threats by making connections to VLSI testing. We cover design-for-trust techniques, such as logic encryption, aging acceleration attacks, and statistical methods that help identify Trojan'ed and counterfeit ICs.**

I. Introduction

Today's System on Chip (SoC) is being incorporated with digital, analog, radio frequency, photonic and other devices [1]. More recently, sensors, actuators, and biochips are also being integrated into these already powerful SoCs. On one hand, SoC integration has been enabled by advances in mixed system integration and the increase in the wafer sizes (currently about 300 mm and projected to be 450mm by 2018 [1]). Consequently, the cost per chip of such SOCs has reduced. On the other hand, support for multiple capabilities and mixed technologies has increased the cost of ownership of advanced foundries. For instance, the cost of owning a foundry will be $5 billion in 2015 [2]. Consequently, only large commercial foundries now manufacture such high performance, mixed system SoCs especially at the advanced technology nodes [3]. Absent the economies of scale, many of the design companies cannot afford owning and acquiring expensive foundries and hence, outsource their design fabrication to these "one-stop-shop" foundries. This globalization of Integrated Circuit (IC) design flow has introduced security vulnerabilities. If a design is fabricated in a foundry that is outside the direct control of the (fabless) design house, reverse engineering, malicious circuit modification, and Intellectual Property (IP) piracy are possible [3]. An attacker, anywhere in this design flow, can reverse engineer the functionality of an IC/IP, and steal and claim ownership of the IP. An untrusted IC foundry may overbuild ICs and sell the excess parts in the gray market. Rogue elements in the foundry may insert malicious circuits (hardware Trojans) into the design without the designer's knowledge [4], [5]. Because of such hardware-based attacks, the semiconductor industry loses $4 billion annually [6].

In this tutorial, we elaborate on these and similar hardware security threats by making connections to VLSI testing. On one hand we show that test techniques may be misused to launch attacks, for example, that accelerate the aging of a circuit, creating reliability problems. On the other hand, we show that concepts, techniques and tools can be adopted from the VLSI testing domain to thwart security threats. In particular, we focus on logic encryption methods that protect against IC/IP piracy and reverse engineering and statistical methods that expose Trojan'ed and counterfeit ICs.

II. Design-for-Trust via Logic Encryption

We first cover a Design-for-Trust (DfTr) technique called Logic Encryption, and show that this technique equipped with fault analysis capabilities can provide a strong defense against reverse engineering, IP piracy, IC overbuilding and Trojans [7], [8]. A logic encryption scheme hardwires designs with built-in keys that are unique to each IC, and ensures that the application of any invalid key on the protected design produces incorrect results. We shed light on the fault activation, propagation, and masking effects in designs and relate them to application of invalid keys to the design, corruption of the outputs of the design and protection of embedded keys.

Logic encryption inserts additional circuit elements into the original design. In order for the design to exhibit its correct functionality (i.e., produce correct outputs), a valid key has to be supplied to the encrypted design. Otherwise, the encrypted design will exhibit a wrong functionality (i.e., produce wrong outputs). The design will be in an encrypted form while it passes through the untrusted design phases, preventing reverse engineering, cloning, trojan insertion and overbuilding. Post-fabrication, the IC is activated by applying the valid key. The secret keys may be stored and embedded in a tamper-evident memory inside the IC to prevent access to an attacker.

We show that testing concepts such as fault activation, propagation, and masking can be utilized to guide the insertion of key gates. This way, a perfect control over the functional corruption due to invalid key application can be achieved [7]. We also show that the same concepts can not only be utilized by an attacker to leak the secret logic encryption key, but also by a designer to guide the insertion of key gates in attaining a hard-to-break logic encryption [8].

III. Attacks for Accelerating Aging

Device aging is an important failure mechanism in nanoscale designs. The circuitry comprising a semiconductor chip degrades over the lifetime of the chip, and ultimately results in chip failure. Circuit degradation is highly influenced by the operating conditions of the circuit

978-1-4673-6377-8/13/$31.00 ©2013 IEEE

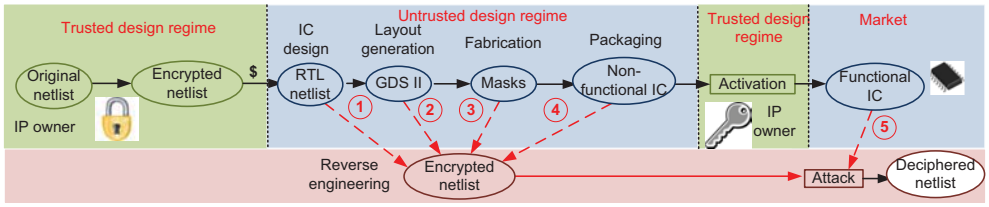


Fig. 1: The logic encryption flow. The design is in the encrypted form in the untrusted design regime. In the untrusted regime, an attacker can obtain the encrypted netlist from (1) the IC design, or by reverse engineering the (2) layout, (3) mask, or (4) a fabricated IC, and (5) the functional IC from the market. Using this attack, the attacker can get a deciphered netlist and make pirated copies if logic encryption is absent or breakable.

including temperature, voltage bias, current density, etc. Rogue designers who are aware of the effect of aging mechanisms on circuit performance, can expedite such aging-related performance degradation and decrease the life time of nanoscale devices intentionally. One possible use case from an attacker' perspective is to accelerate the failure of consumer electronics devices when the device is still under replacement warranty.

Negative Bias Temperature Instability (NBTI) is one of the major threats to the reliability of nanoscale designs. Being aware of direct impacts of NBTI on circuit performance over time, an attacker can benefit from NBTI effects and expedite aging-related performance degradation. In this research, we will describe an NBTI-related aging attack, and deal with countermeasures to prevent this attack.

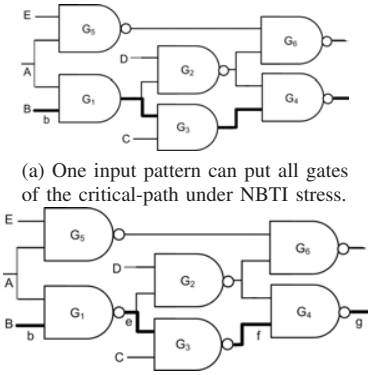
A. NBTI Aging

NBTI occurs when traps are generated at the Si-SiO2 interface when a negative voltage is applied to a PMOS device [9]. NBTI increases the magnitude of threshold voltage (V_{th}) of the PMOS transistor under stress and reduces the drain current through it and hence degrades the delay through the distressed PMOS transistor. At the circuit level this manifests as circuit timing and functional failures. To mitigate the performance degradation of NBTI and increase the reliability of circuits, several methods such as guard-banding, gate-sizing, voltage-tuning, and body-biasing have been proposed in literature [10], [11], [12], [13]. However, even when designs are equipped with these aging-mitigation schemes, aging attacks can be quite effective and adversely affect the performance of nanoscale devices.

B. Accelerating NBTI Aging

NBTI occurs when a PMOS device is on. Since, threshold voltage of a PMOS transistor is a function of the duration that the transistor is on, by keeping a PMOS transistor turned on its performance can be degraded. Accordingly, by keeping all the PMOS transistors residing in the longest path of a circuit turned on, NBTI aging can be accelerated.

Fig. 2a shows a sample circuit. In this circuit, path P_1 (including G_1, G_3, and G_4) is the longest path of the circuit. By holding the primary inputs constant at V_1="10111", the PMOS transistors residing in the longest path of the circuit



(a) One input pattern can put all gates of the critical-path under NBTI stress.

(b) Two input patterns can put all gates of the critical-path under NBTI stress.

Fig. 2: Sample circuits (longest path depicted in bold).

are kept 'on' and accordingly all the gates residing in the critical path of the circuit experience NBTI aging sooner than expected. Considering the topology of a circuit and type of the gates residing in the critical path of the circuit, it is not always possible to put all the gates residing in the critical path of the circuit under NBTI stress using one single pattern. For example, consider the circuit shown in Fig. 2b. In this circuit, no single pattern can accelerate NBTI aging in G_1, G_3, and G_4, simultaneously. However, holding the primary inputs constant at V_1="10111" accelerates the aging of G_1 and G_4, while holding the primary inputs constant at V_2="11111" accelerates the aging of G_3. In this research, we aim at finding minimum number of patterns using which we get maximum NBTI-aging acceleration.

Generating NBTI-acceleration patterns: To be able to put all the gates in the critical path of a circuit under NBTI stress using minimum number of vectors, one simple solution is using path-delay testing patterns. In practice, to ensure that the propagation delay of each paths in a circuit is less than the pre-defined clock period, each path should be tested individually. To test a circuit path for a path-delay fault, two test patterns are required. First, pattern V_1 is applied to the circuit. Then, when the signals in the circuit have stabilized, the second pattern (V_2) is applied to launch a transition from the input along the tested path to the output of the path. By sampling the path output, correct/faulty behavior of the path is identified [14][15].

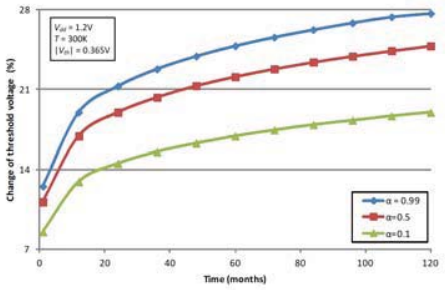


Fig. 3: Percentage change in V_{th} of a PMOS device as a function of time for different values of α.

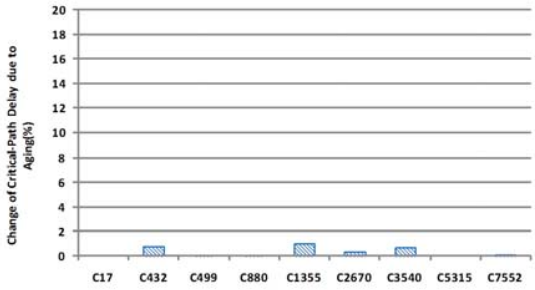
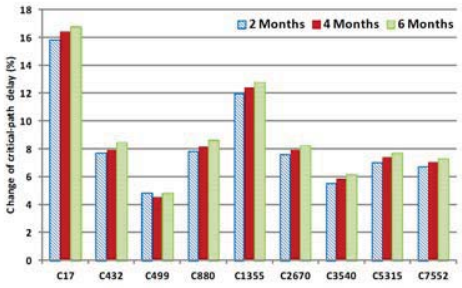
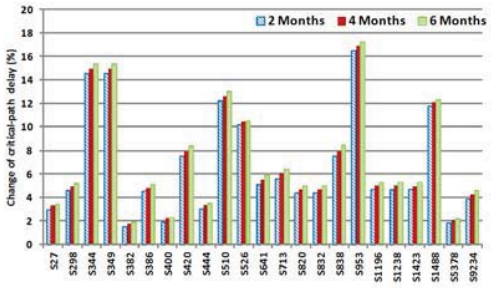


Fig. 4: Percentage change in critical-path delay of benchmark circuits applying random patterns for 4 months.



(a) ISCAS'85 benchmarks



(b) ISCAS'89 benchmarks

Fig. 5: Percentage change in critical-path delay of benchmark circuits as a function of attack duration.

To accelerate NBTI-aging in a given circuit, we leverage ATPG tools to generate path-delay testing patterns targeting the critical path of that circuit. By applying the generated pair of patterns to the circuit, a transition is launched through that path and therefore all the lines in that path get a "0" either when the first or when the second pattern is applied. Accordingly, every PMOS transistor fed by those lines is turned on by continuously applying one of the two patterns. The time duration of applying the generated patterns to the circuit as well as the type of gates in the targeted path determines the magnitude of aging.

To apply NBTI-acceleration patterns, first the circuit is switched to the shift-mode where the first pattern (V_1) is applied to the circuit. Then the shift clock will be frozen for time duration of D_1 after which pattern V_2 is applied to the circuit and the shift-clock will be frozen for time duration of D_2.

C. Attack Model

In this threat, the attacker obtains the netlist of the target hardware either from a rogue insider in the design house or by reverse engineering it. Then he uses VLSI testing tools to determine the input patterns that accelerate aging of the hardware. Then, the attacker provides a software program using which generated patterns are applied to the target hardware in the operating system mode.

D. Evaluating NBTI Aging Effects

To evaluate the effect of NBTI aging on the threshold voltage of PMOS devices, we use the simplified model presented in [16], where for the 65nm technology, V_{dd} = 1.2V and T = 300K, ΔV_{th} is expressed using Equation 1. In this equation, α shows the fraction of time the transistor is under NBTI stress. Fig.3 quantifies the effect of NBTI stress on a PMOS device over time.

$$\Delta V_{th} = b \cdot \alpha^n \cdot t^n, b = 3.9 \times 10^{-3} V/s^6, n = 0.16 \quad (1)$$

E. Experimental Results

To show the effectiveness of our method in degrading the performance of each circuit, we extracted the magnitude of performance degradation of ISCAS benchmarks when NBTI-accelerating patterns are applied for 2, 4 or 6 months to each circuit (each pattern for 50% of the duration). The results are shown in Fig. 5. As depicted in this figure, on average the performance of ISCAS'85 benchmarks is degraded by 8.3% after 2 months, by 8.6% after 4 months, and by 9% after 6 months. Similarly, the performance of ISCAS'89 circuits degraded by 6.6%, 7%, and 7.3% after 2, 4 and 6 months, respectively. Since, NAND gates are highly susceptible to NBTI aging and experience more performance degradation compared to other logic gates, NAND-dominated circuits experience more performance degradation compared to other circuits.

We also compared the efficiency of NBTI-acceleration patterns to randomly generated patterns in degrading the performance of each circuit. Accordingly, we generated a pair of random patterns for all ISCAS benchmarks and continuously applied them for four months (each pattern for two months). We repeated this experiment for five times. As shown in Fig. 4, by applying randomly generated patterns

to a circuit for four months, on average, the performance of ISCAS'85 benchmarks degraded by 0.23% and performance of ISCAS'89 benchmarks degraded by an insignificant, near 0%, amount.

Note that in these experiments we only considered the NBTI degradation effect and didn't consider the recovery phase that occurs when an under-stress PMOS transistor turns off. We expect a slight change in the NBTI effect evaluation when the recovery phase is considered.

IV. Statistical Detection of Hardware Trojans

A. Hardware Trojans

The problem of hardware Trojans in manufactured ICs emerged recently and has already garnered interest not only in academia but also in governmental agencies and industry. Hardware Trojans are malicious modification that can be exploited by an adversary to cause incorrect results, steal sensitive data, or incapacitate a chip. Traditional test methods fall short in detecting such Trojans, as they are only geared towards identifying modeled defects and, therefore, cannot reveal unmodeled malicious inclusions.

B. Side-Channel Based Statistical Detection

Among the various approaches proposed by researchers, the use of statistical analysis of side-channel measurements to detect Trojan-infested circuits has been the most promising one. The key idea is that, while parametric deviations caused by inserted hardware Trojans might be small and may very well be hidden in the design margins, they are systematic and therefore a statistical analysis approach will be able to pick up the additional structure they bring to the fingerprint. In the digital domain, the first works along this line were presented in [17], where the global power consumption is chosen to construct side-channel fingerprints and in [18], where fingerprints for path delays are built. After the proposals of [17], [18], many researchers in the hardware trust area picked up this idea and pursued it further by using different side-channel measurements and/or their combinations which include power supply transient signals [19], [20], leakage current [21], regional supply current [22], and multi-parameter combinations [23].

While most of these methods are for digital circuits, the analog domain is even richer in parametric measurements, making statistical analysis methods attractive in that domain. Furthermore, the analog/RF domain is also an obvious target for attacks, since the wireless communication of these chips with the environment simplifies the process of staging the attack without obtaining physical access to the I/O of the chip. The first investigation of the side-channel fingerprinting method in detecting hardware Trojans in these circuits appeared in [24]. The experimentation vehicle used in that study is a mixed-signal wireless cryptographic integrated circuit, capable of encrypting and broadcasting data, which can be used in secure data transmission over open channels.

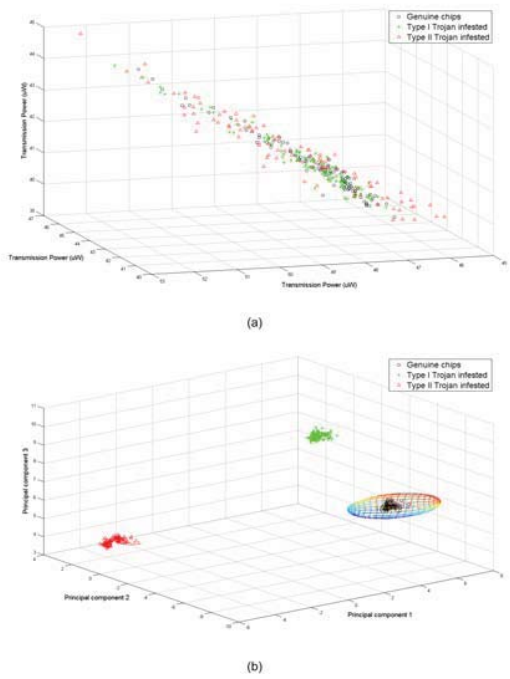


Fig. 6: (a) Projection of genuine and Trojan-infested chip populations on three out of six transmission power measurements, (b) Projection of genuine and Trojan-infested chip populations on three principal components of six transmission power measurements.

The digital part includes a pipelined Digital Encryption Standard (DES) core, an output buffer and a serializer. The analog part is an Ultra-Wide-Band (UWB) transmitter. Two hardware Trojans were also designed, capable of leaking the encryption key by hiding it in the wireless transmission amplitude (Type-I) or frequency (Type-II) margins allowed due to process variations; thus, they ensure that the circuit continues to comply to all of its functional specifications.

In order to construct the side-channel fingerprints, measurements of the total transmission power for broadcasting one block of data (i.e. 64-bits) are collected. Using Monte Carlo simulation with 5% process variations, 100 Type-I Trojan-infested, 100 Type-II Trojan-infested, and 200 Trojan-free circuit instances are generated from which we measure the total transmission power when transmitting each of six randomly selected blocks (the same for all circuits). All six measurements for all genuine and all Trojan-infested chips are within the acceptable specification range. Even when we project the three chip populations on the six-dimensional space of these measurements, we cannot distinguish them since they fall upon each other. Figure 6(a) shows a projection of the three populations on three of these dimensions. Evidently, separating the genuine from the Trojan-infested populations in this space is not possible. The situation is similar for any other subset of three measurements.

However, running a Principal Component Analysis (PCA) [25] on these measurements reveals that the structure of the genuine chip data is different than the structure of the Trojan-infested chip data. Figure 6(b) shows a projection of the three populations on the three principal components of the data, clearly revealing that they are separable in this space. Therefore, we can define the trusted boundary as a simple minimum volume enclosing ellipsoid (MVEE [26]) which encompasses the genuine population. Then, we can discard as suspicious any chip whose footprint on the space of the selected three principal components does not fall within the trusted boundary. In our example, this method detects all Type-I and Type-II Trojan-infested chips without inadvertently discarding any genuine chips.

V. Counterfeit ICs: Statistics to the Rescue

A. Counterfeit ICs

Today's IC supply chain has grown very complex and globalized, with constituents of an electronics production flow scattered across the globe and with parts typically coming from multiple suppliers. In such an environment, is not always feasible to guarantee that each part provided by the suppliers is a legitimate, brand-new device. For example, some ICs provided by dubious suppliers, could potentially be "recycled" from used or defective circuit boards. Such ICs, constitute one type of counterfeits[1] and are the topic of this section. While such ICs can initially function properly, they have been subjected to various aging phenomena and are, therefore, a reliability risk due to their reduced expected lifetime. Such counterfeit ICs have turned up in many industrial sectors, including computers, telecommunications, automotive electronics, and even military systems. Several practices exist to identify counterfeit ICs, including visual inspection [28] or part authentication tools [29] which consist of providing an encrypted number for each device by an RFID tag in production. However, the time and the cost required for applying these methods is prohibitive due to the economic pressures in a modern electronic development and fabrication flow. Accordingly, there is a pressing need to develop low-cost methods for identifying counterfeit ICs.

B. Statistical Detection

To this end, we describe a method for detecting counterfeit ICs through the use of Support Vector Machines (SVMs). Specifically, we train a one-class classifier to distinguish aged devices from brand-new ones, using measurements from production Early Failure Rate (EFR) analysis. Such measurements are required prior to releasing most products, hence no additional cost is incurred for counterfeit IC detection. Figure 7 illustrates the proposed method for identifying a counterfeit IC in a two-dimensional parametric measurement space. The first step involves collection of a set of parametric measurements, which are taken on a set of brand-new devices (which are subject to process variations), obtained from a trustworthy provider. These measurements are used to train a one-class SVM in order to allocate a non-linear separation boundary in the space of parametric measurements, to separate brand new devices and counterfeit devices. This approach is inspired by and resembles closely a machine learning-based analog/RF IC test method [30].

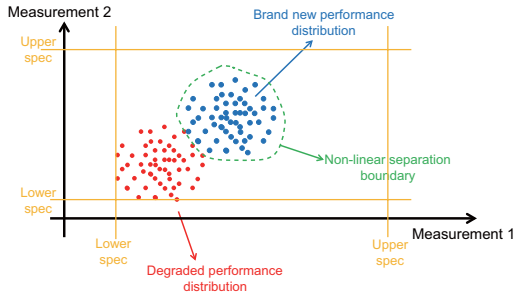


Fig. 7: Proposed counterfeit IC identification approach.

C. Experimental Results

We demonstrate effectiveness of this method on a Digital Signal Processor (DSP) involving 49 parametric test measurements performed on 313 devices randomly chosen from different lots in production. These measurements are taken at 5 different time points for the same devices during burn-in test, in order to mimic the impact of aging degradation over time: t_0, t_1, t_2, t_3, t_4. Devices at $t = t_0$ are referred to as brand new, while devices at $t = t_i, i > 0$ are referred to as counterfeit. Since we have a relatively high data dimensionality d for this case study (d = 49), we first perform a Principal Component Analysis (PCA) in order to map the original 49 measurements onto vectors in a lower dimensional space with cardinality d' <49. We maintain the structure of the data while keeping only 9 principal components, i.e. d' = 9. Figures 8 and 9 show the projection of devices at $t = t_0, t_1$ and $t = t_0, t_4$, respectively, onto the first three principal components. As can be clearly observed, performance degradation caused by aging mechanisms is accelerated during the burn-in test. Indeed, an SVM trained with half of the devices at time $t = t_0$, achieves 100% correct group classification rate for classifying devices at $t = t_0, \ldots, t_4$, respectively, showing the excellent capability of detecting counterfeit devices using this approach [31].

VI. Summary

The various aspects of ensuring security and trustworthiness in integrated circuits call for solutions which, in many ways, borrow ideas from and resemble similar methods developed by the VLSI test community. As elucidated by this tutorial, problems such as IP/IC theft, accelerated

[1]Other types of counterfeit ICs include chips from overproduction by an untrusted foundry, reverse-engineered ICs, and fakes. According to [27], legitimate electronics companies miss out on about $100 billion of global revenue every year because of counterfeiting.

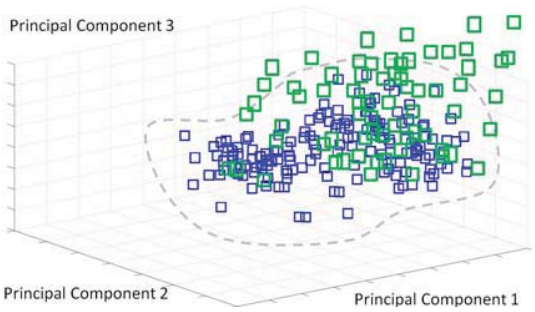


Fig. 8: Projection of devices at $t = t_0, t_1$, shown by blue and green squares, respectively.

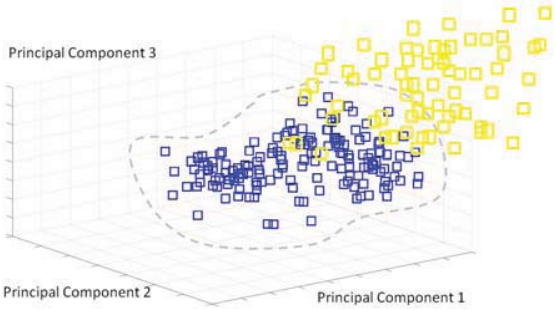


Fig. 9: Projection of devices at $t = t_0, t_4$, shown by blue and yellow squares, respectively.

transistor aging-based attacks, Trojan'ed ICs, and counterfeit ICs, among others, can all be approached by leveraging concepts from design-for-test, ATPG, reliability analysis, and statistical test methods, which the test community has long been using extensively. By pointing out the similarities, this tutorials seeks to encourage more test researchers to engage in this contemporary and extremely important problem of hardware security.

ACKNOWLEDGEMENTS

The work carried out at NYU is funded in parts by NSF awards 0966187 and 1059328, by the NYU Center for Research in Information Security Studies and Privacy (CRISSP) and by AFRL under contract No. FA8750-11-2-0274. The work carried out at UT Dallas is funded in parts by NSF award 1149465 and by ARO under contract No. W911NF-12-1-0091. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF, CRISSP, AFRL or ARO.

REFERENCES

[1] "International Technology Roadmap for Semiconductors," http://www.itrs.net/Links/2011ITRS/Home2011.htm.

[2] DIGITIMES Research, "Trends in the global IC design service market," http://www.digitimes.com/Reports/Report.asp?datepublish=2012/3/13\&pages=RS\&seq=400\&read=toc.

[3] Intelligence Advanced Research Projects Activity, "Trusted Integrated Circuits Program," https://www.fbo.gov/utils/view?id=b8be3d2c5d5babbdffc6975c370247a6.

[4] J. Roy, F. Koushanfar, and I. Markov, "EPIC: Ending Piracy of Integrated Circuits," *Design, Automation and Test in Europe (DATE)*, 2008 pp. 1069–1074, 2008.

[5] R. Chakraborty and S. Bhunia, "HARPOON: An Obfuscation-Based SoC Design Methodology for Hardware Protection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 10, pp. 1493–1502, 2009.

[6] SEMI, "Innovation is at risk as semiconductor equipment and materials industry loses up to $4 billion annually due to IP infringement," www.semi.org/en/Press/P043775, 2008.

[7] J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri, "Logic encryption: A fault analysis perspective," in *Design, Automation and Test in Europe (DATE)*, 2012, pp. 953 –958.

[8] ——, "Security analysis of logic obfuscation," in *Design Automation Conference (DAC)*, 2012, pp. 83 –89.

[9] S. Bhardwaj, W. Wang, R. Vattikonda, Y. Cao, and S. Vrudhula, "Predictive modeling of the NBTI effect for reliable design," in *Custom Integrated Circuits Conference (CICC)*, Sept. 2006, pp. 189–192.

[10] J. Abella, X. Vera, and A. Gonzalez, "Penelope: the NBTI-aware processor," in *International Symposium on Microarchitecture (MICRO)*, 2007, pp. 85–96.

[11] R. Vattikonda, W. Wang, and Y. Cao, "Modeling and minimization of pmos nbti effect for robust nanometer design," in *Design Automation Conference (DAC)*, 2006, pp. 1047–1052.

[12] D. R. Bild, G. E. Bok, and R. P. Dick, "Minimization of nbti performance degradation using internal node control," in *Design Automation and Test in Europe (DATE)*, 2009, pp. 148–153.

[13] Y. Lee and T. Kim, "A fine-grained technique of nbti-aware voltage scaling and body biasing for standard cell based designs," in *Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2011, pp. 603–608.

[14] C. J. Lin and S. Reddy, "On delay fault testing in logic circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 6, no. 5, pp. 694–703, september 1987.

[15] S. Bose and V. Agrawal, "Delay test quality evaluation using bounded gate delays," in *VLSI Test Symposium (VTS)*, 2007, pp. 23–28.

[16] W. Wang, Z. Wei, S. Yang, and Y. Cao, "An efficient method to identify critical gates under circuit aging," in *International Conference on Computer-Aided Design (ICCAD)*, 2007, pp. 735–740.

[17] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, "Trojan Detection using IC Fingerprinting," *IEEE Symposium on Security and Privacy (S&P*, pp. 296–310, May 2007.

[18] Y. Jin and Y. Makris, "Hardware Trojan detection using path delay fingerprint," *IEEE International Workshop on Hardware-Oriented Security and Trust (HOST)*, pp. 51–57, Jun 2008.

[19] R. M. Rad, X. Wang, M. Tehranipoor, and J. Plusquellic, "Power supply signal calibration techniques for improving detection resolution to hardware Trojans," in *International Conference on Computer-Aided Design (ICCAD)*, 2008, pp. 632–639.

[20] R. Rad, J. Plusquellic, and M. Tehranipoor, "Sensitivity analysis to hardware Trojans using power supply transient signals," in *IEEE International Workshop on Hardware-Oriented Security and Trust (HOST)*, 2008, pp. 3–7.

[21] C. Lamech, J. Aarestad, J. Plusquellic, R. Rad, and K. Agarwal, "REBEL and TDC: two embedded test structures for on-chip measurements of within-die path delay variations," in *International Conference on Computer-Aided Design (ICCAD)*, 2011, pp. 170–177.

[22] D. Du, S. Narasimhan, R. Chakraborty, and S. Bhunia, "Self-referencing: A scalable side-channel approach for hardware Trojan detection," in *Cryptographic Hardware and Embedded Systems, CHES 2010*, 2010, pp. 173–187.

[23] S. Narasimhan, D. Du, R. Chakraborty, S. Paul, F. Wolff, C. Papachristou, K. Roy, and S. Bhunia, "Multiple-parameter side-channel analysis: A non-invasive hardware trojan detection approach," in *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, 2010, pp. 13–18.

[24] Y. Jin and Y. Makris, "Hardware Trojans in wireless cryptographic ICs," *IEEE Design and Test of Computers*, vol. 27, pp. 26–35, 2010.

[25] I. T. Joliffe, *Principal Component Analysis*. Springer-Verlag, 1986.

[26] N. Moshtagh, "Minimum volume enclosing ellipsoid," in *GRASP Laboratory, University of Pennsylvania, http://www.seas.upenn.edu/ñima/papers/Mim_vol_ellipse.pdf*, 2005.

[27] M. Pecht and S. Tiku, "Bogus: electronic manufacturing and consumers confront a rising tide of counterfeit electronics," *IEEE Spectrum*, vol. 43, no. 5, pp. 37–46, 2006.

[28] "Detection of counterfeit electronic components," *http://www.aeri.com/detection-of-counterfeit.asp*.

[29] K. Chatterjee and D. Das, "Semiconductor manufacturers' efforts to improve trust in the electronic part supply chain," *IEEE Transactions on Component Packaging Technology.*, vol. 30, no. 3, pp. 547–549, 2007.

[30] H.-G. Stratigopoulos and Y. Makris, "Error moderation in low-cost machine-learning-based Analog/RF testing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 2, pp. 339–351, 2008.

[31] K.Huang, J. Carulli, and Y. Makris, "Parametric counterfeit ic detection via support vector machines," in *Proc. IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFTS)*, 2012, pp. 7–12.