When Capacitors Attack: Formal Method Driven Design and Detection of Charge-Domain Trojans

Xiaolong Guo*, Huifeng Zhu†, Yier Jin* and Xuan Zhang†
*Department of Electrical and Computer Engineering, University of Florida
†Department of Electrical and Systems Engineering, Washington University in St. Louis
guoxiaolong@ufl.edu, zhuhuifeng@wustl.edu, yier.jin@ece.ufl.edu, xuan.zhang@wustl.edu

***Abstract*—The rapid growth and globalization of the integrated circuit (IC) industry put the threat of hardware Trojans (HTs) front and center among all security concerns in the IC supply chain. Current Trojan detection approaches always assume HTs are composed of digital circuits. However, recent demonstrations of analog attacks, such as A2 and Rowhammer, invalidate the digital assumption in previous HT detection or testing methods. At the system level, attackers can utilize the analog properties of the underlying circuits such as charge-sharing and capacitive coupling effects to create information leakage paths. These new capacitor-based vulnerabilities are rarely covered in digital testings. To address these stealthy yet harmful threats, we identify a large class of such capacitor-enabled attacks and define them as charge-domain Trojans. We are able to abstract the detailed charge-domain models for these Trojans and expose the circuit-level properties that critically contribute to their information leakage paths. Aided by the abstract models, an information flow tracking (IFT) based solution is developed to detect charge-domain leakage paths and then identify the charge-domain Trojans/vulnerabilities. Our proposed method is validated on an experimental RISC microcontroller design injected with different variants of charge-domain Trojans. We demonstrate that successful detection can be accomplished with an automatic tool which realizes the IFT-based solution.**

I. Introduction

The exponential growth of the integrated circuit (IC) industry results in rapid globalization of its supply chain. Since a complicated IC design often involves numerous IP suppliers, fabrication foundries, and testing facilities spanning multiple continents, it is extremely challenging, if not outright impossible, to track the source of every component and secure the entire supply chain. The sophistication of today's IC development process gives rise to the increasing threats of hardware Trojans (HTs). Ever since its first conceptualization in 2007 [1], the field of HT research has experienced notable growth and various HT attack and defense mechanisms have been exploited, demonstrated, and refined. Until recently, most HTs are deployed as digital circuits, and their detection approaches also follow the same assumption. However, seminal works presented recently, such as A2 [2] and Rowhammer [3], demonstrate that analog circuits or analog properties of digital circuits can be leveraged to launch stealthy HT attacks.

Unfortunately, existing HT countermeasures targeting the digital circuits may not be applied to these analog Trojans, because the analog-style behaviors of the circuits are abstracted away during the verification and checking stage of the IC design process. Although testing methods like R2D2 can detect A2 Trojans associated with specific frequencies and registers [4], sophisticated attackers can bypass the detection by modifying the Trigger frequency and the insertion position. VeriCoq IFT represents another approach to detecting analog-style HTs [5]. It implements analog information flow tracking (IFT) at the transistor level and demonstrates a detection of an electrocardiogram (ECG) signal leakage. Yet, VeriCoq IFT suffers from a high false positive rate because it fails to take the features of the analog attack into account. Although exhaustive testing of all the scenarios involving the analog circuits could in theory expose the HT trigger events, the probability to activate such rare events among billions of logic gates is close to zero, not to mention that it would significantly increase the time-to-market (TTM) and hurt profitability and competitiveness. Therefore, the threats of analog Trojans call for an effective and low-cost detection method that is fundamentally different from the existing digital domain approaches.

In this paper, we investigate a systematic method to detect a large class of analog Trojans that act in the charge domain. These analog threats, like A2 Trojans and Rowhammer attacks, create information leakage paths through electrical charge transfer. They utilize subtle analog behaviors of low-level circuits and thus cannot be exposed by HT countermeasures in the digital domain. Adversaries can stealthily insert malicious additions or make use of existing vulnerabilities in the circuits. The key to develop countermeasures for analog attacks is the abstraction of analog/mixed-signal behaviors that can provide an effective measurement metric in detection. Specifically, we identify a charge-domain metric to describe a general form of information leakage paths that are facilitated by capacitor circuits. Depending on whether the capacitors are intentional or parasitic, these charge-domain Trojans can be further classified as charge-sharing and capacitive-coupling. We delve more deeply into the the charge-sharing Trojans where switched-capacitor circuits are intentionally inserted to enable charge-domain information leaks. Aided by the new abstraction model, we are able to infer variants of charge-sharing Trojans, among which A2 is but one example. The main contributions of this paper are listed as follows:

- We broadly define charge-domain Trojans from a systematic perspective and provide abstractions for their subsets—charge-sharing and capacitive-coupling leakage paths, accordingly. This work, for the first time, identifies a wide existing class of analog/mixed-signal threats. Our charge-domain abstraction presents an effective metric for researchers to study and analyze such stealthy attacks.
- We leverage the charge-domain leakage path model and develop an IFT based detection scheme for analog/mixed-signal Trojans. Compared with previous digital-only IFT methods, we design the information flow policy with the consideration of fine-grain charge-domain behaviors. To our knowledge, it is the first IFT solution that can efficiently detect threats from analog/mixed-signal circuits.

978-3-9819263-2-3/DATE19/©2019 EDAA

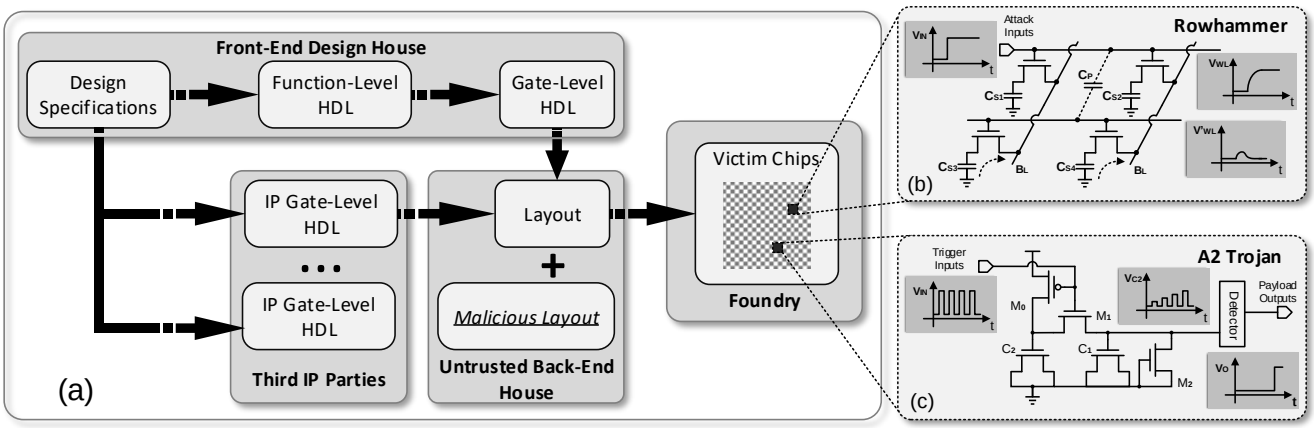

Figure 1: Attack model and examples of analog hardware Trojans.

- An automated tool is developed to demonstrate the effectiveness of the information flow tracking on analog Trojans detection in the benchmark which leverages customized information flow policies.

II. Analog Trojans and Attack Model

A. Analog Hardware Trojans

Recently, analog HTs have attracted increasing attention because of their immunity to traditional digital-domain HTs detection and testing methods as well as their heightened emerging threats to modern systems. While some analog Trojans exploit the sensitivity of analog/mixed-signal circuits as compared to their digital counterparts to create reliability issues [6], more lethal forms of analog attacks often hide inside common ubiquitous digital blocks. Among them are recently-demonstrated charge-domain Trojans, such as A2 Trojans [2] and Rowhammer attacks [3].

A2 Trojan, as illustrated in Figure 1(c), is a type of charge-domain Trojans with small footprint and minimal power impact. Attackers employ a toggling register as trigger input to periodically charge C_2 and then redistribute the charges among C_1 and C_2. The result is a steadily rising voltage across C_1. Once the trigger frequency increases above a threshold, the Trojan payload activates as C_1's voltage crosses the detector threshold. Rowhammer is another type of analog attacks that is widely existed in modern DRAMs, as shown in Figure 1(b). When adversaries repeatedly toggle a wordline, the parasitic capacitance between wordlines causes charge disturbances on the adjacent rows by accelerating the charge leakage rate of the memory cells connected to the victim rows. If an effected cell loses too much charges before it is refreshed to the original value, a memory error occurs.

In this paper, we systematically define a broad class of these analog attacks as charge-domain Trojans and elaborate its detailed classification and usage in Section III.

B. Attack Model

We assume that the adversary can employ the existing analog properties or implement malicious functions at the back-end stage or fabrication stage in order to form an information leakage path, as shown in Figure 1(a). At the model abstraction step, we target both charge-sharing and capacitive-coupling attacks as defined in Section III-A. We are able to derive the abstracted models of sneaky information leakage paths used by these attacks that can be leveraged as effective features by the detection methods. At the experiment step, however, we focus more specifically on charge-sharing Trojans and refine our attack model at the back-end or fabrication stage. After the gate-level design, the front-end design house delivers the IPs to an untrusted back-end house. Similar to previous work [2], attacks are performed after the place&route process. By directly modifying the layout in the Graphic Database System II (GDSII) file, the adversary can insert malicious analog components, and manipulate the corresponding post-layout chip-level netlist to trick the Layout v.s. Schematic (LVS) verification as well. Compared with the original RTL level netlist from the design house, the post-layout system-level netlist contains many additional components, such as clock buffers, decoupling capacitors, fillers, and dummy cells, among which malicious analog components can be disguised. Although we also sketch out a frequency-based IFT methods to automatically detect the capacitive-coupling threats, due to space limit, we only elaborate on detecting charge-sharing Trojans in the experiments.

III. Abstraction of Charge-Domain Trojans

A. Charge Domain Modeling

In our definition, charge-domain Trojans belong to a large class of analog threats whose attacks are launched through malicious and deliberate electrical charge transfers and/or redistribution. They are prevalent in modern digital IC systems, because at a fundamental level, each digital bit is stored as charges across a capacitor and the operation of charge-domain Trojans weaponizes the capacitive effects by disturbing the normal charge level of critical nodes. They are often deployed with synergistic hardware and software coordination. A general form of charge-domain Trojans can be described as the electrical charge accumulation across an essential capacitor. Each trigger activity i results in charge disturbance of $\Delta Q(i)$. The charge disturbances accumulate over many iteration of trigger events until they eventually reach a critical value (Q_{cr}) to enable the payload circuit and implement the attack, as captured by the following expression:

$$|\sum_{i=0}^{N} \Delta Q(i)| > Q_{cr} \quad (1)$$

This charge-domain formulation captures a wide range of analog attacks and can be used to describe existing practical analog Trojans. We further divide them into two subclasses—charge-sharing and capacitive-coupling Trojans, according to their distinctive attack models. The former involves insertion or modification of the physical design by the adversaries to enable deliberate charge sharing behaviors, whereas the latter relies on capacitive coupling associated with the parasitics intrinsicly residing in the original digital circuits and requires no hardware modifications.

B. Model of Capacitive-coupling Trojans

One practical example of the capacitive-coupling Trojans is the Rowhammer attacks. As shown in Figure 1(b), when a wordline, noted as trigger wordline, is activated, due to the parasitic capacitor between wordlines, the adjacent wordline, noted as victim wordline, experiences undesired voltage fluctuation that affects the access transistor and accelerates charge leakage at the store capacitor (C_S in a DRAM cell). To apply our general form Equation (1) to Rowhammer attacks, we can identify C_S as the capacitor of interest. Every time the trigger wordline is activated, $\Delta Q(i) = kC_P R_{WL} V_{WL} \mathcal{F}$, where C_P is the parasitic capacitance between the two wordlines, R_{WL} and V_{WL} are the resistance and the voltage of the wordline. Coefficient k is introduced to describe the degree of the capacitive coupling effects, and the stochastic function $\mathcal{F}$ is used to describe the probabilistic charges leakage when the access transistor works in the sub-threshold region. After each DRAM refresh, C_S is initially charged to V_{DD} and the accumulated charge leakage could cause the voltage to drop below the memory threshold (V_{th}) resulting in an erroneous bit flip. Therefore, Equation (1) can be rewritten as:

$$\sum_{i=0}^{N} (kC_P R_W V_{DD} \mathcal{F}_i) > (V_{DD} - V_{th}) C_S \tag{2}$$

where Q_{cr} is determined by $(V_{DD} - V_{th})C_S$. Note that Equation (2) correctly captures the main underlying mechanism of Rowhammer, as it shows that if the trigger wordline is activated many times (sufficient large N) during the interval between two refreshes, the cells on victim wordline may incur charge leakage beyond Q_{cr} and experience disturbance errors.

Due to its parasitic nature, sneaky paths used by capacitive-coupling Trojans are omnipresent in digital designs and can only be detected by analyzing the extracted netlist from a layout. An efficient method is needed to sort through all the possible leakage paths enabled by parasitic capacitance, identify effective attack mechanisms with high probability, and rule out false positive instances. In the case of Rowhammer, we believe it can be achieved by properly formulating $\mathcal{F}$ as a stochastic function of device-level process variation and estimating the feasible range of N as a statistical expectation. Unlike previous memory behavioral Rowhammer models which rely solely on empirical data to determine the activation threshold [7], our abstraction reflects its physical origin and has the potential to discover new unobserved security phenomenon. However, this work focuses on the definition and model abstraction of capacitive-coupling Trojans and leaves the implementation of detection method for future work.

C. Model of Charge-Sharing Trojans

According to the attack model distinctions, charge-sharing Trojans refer to the subset of charge-domain attacks where dedicated analog circuits, as shown in Figure 2(a), need to be inserted to the physical design during back-end or fabrication stages to intentionally create the sneaky paths. The malicious circuits that enable charge sharing often fall into the category of switched capacitor circuits, and one popular manifestation of charge-boosting Trojans is the switched-capacitor based A2 Trojan. To derive the A2 attack mechanism following our general charge-domain formulation, C_1 is identified as the essential storage capacitor and after each trigger cycle, the charges across C_1 and C_2 redistribute. During the positive phase of a trigger cycle, S_1 is closed and S_2 is open. The sampling capacitor C_2 is charged to V_{DD}. In the negative phase, C_1 and C_2 are shorted together to boost the charges across C_1. So Equation (1) can be rewritten as:

$$\sum_{i=1}^{N} \frac{C_1 C_2 V_{DD} - C_2 Q_1(i-1) - C_1 Q_{leak}}{C_1 + C_2} > C_1 V_{th} \tag{3}$$

where $Q_1(i-1)$ is the original charges of C_1 before the charge sharing (specifically $Q_1(0) = 0$), Q_{leak} is the charge leakage of C_1 during each cycle, and Q_{cr} is determined by the threshold voltage (V_{th}) of payload circuit.

D. Variants of Charge-Sharing Trojans

In addition to the circuit realization used in the original A2 Trojan, three alternative switched-capacitor topologies are commonly employed, as shown in Figure 2. It turns out that among all the topologies, the one in Figure 2(b) is the only one that can increase charges to satisfy the critical charge accumulation requirement expressed in Equation (1), as the other three implementations invariably decrease the charges and therefore are unable to overcome the intrinsic leakage currents. This observation allows us to concentrate on this particular type of topology for charge-sharing Trojans detection. Despite the limitation of the switched-capacitor topology, charge-sharing Trojans can still take many circuit forms with varying trigger patterns. For example, besides the original trigger pattern of the A2 Trojan using a pair of complementary MOS switches as in Figure 2(f), attackers can adopt more elaborate MOS switch trigger signals, as shown in Figure 2(g)-(l). These variants resemble standard digital cells, such as NAND, NOR, AND-OR-INV (AOI), and OR-AND-INV (OAI), hence can be designed to easily hide in regular digital circuits. However, since the formulation of charge-domain model is based on abstracting the charge-storing capacitor of interest, all these possible variants can be effectively represented no matter how sophisticated their trigger patterns are.

Taking the practical implementations into consideration, the discharge by leakage current during one cycle is I_{leak}/f_{SW}, where I_{leak} is the leakage current of C_1. The switched-capacitor circuit reaches the stable state when $\Delta Q(i) = 0$. We can derive the range of trigger frequency f_{SW} that is required to efficiently trigger the payload circuits:

$$f_{SW} > \frac{1}{V_{DD} - V_{th}} \frac{I_{leak}}{C_2} \tag{4}$$

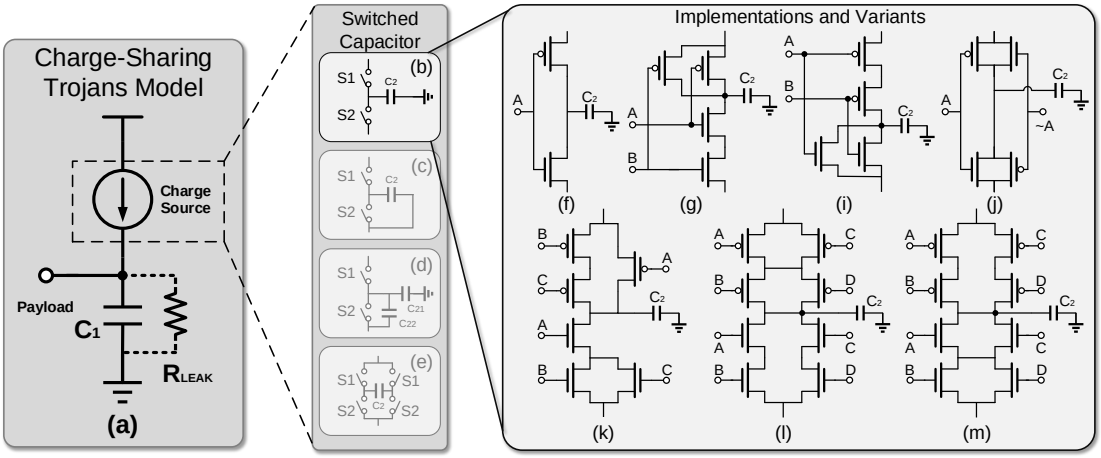

Figure 2: The model, variants and implementations of charge-sharing Trojans.

Since MOS capacitors are most commonly implemented in modern IC, the gate oxide leakage current dominates the leakage and the gate oxide parasitic capacitor dominates C_2. Hence both can be approximated:

$$I_{leak} \approx I_g W_1 L_1, C_2 \approx C_{ox} W_2 L_2 \tag{5}$$

where I_g is the unit gate oxide leakage current, C_{ox} is unit gate oxide capacitance, and W and L are the width and length of MOS capacitors. Therefore the final expression of f_{SW} as a function of process and design parameters becomes:

$$f_{SW} > \frac{W_1 L_1}{W_2 L_2} \frac{I_g}{C_{ox}} \frac{1}{V_{DD} - V_{th}} \tag{6}$$

In short channel processes, I_g is large due to the thin gate oxide. So the size of C_2 must be sufficiently large such that the minimum required trigger frequency falls into a feasible range. Conversely, in long channel processes, I_g is relatively low and the triggers are too frequently. To avoid false triggering, the size of C_1 must be large to set a higher minimum trigger frequency. In both cases, the specific size requirement of the switched capacitors become a critical identifiable feature of potential charge-sharing Trojans.

IV. IFT based Secure Solution

Information flow tracking (IFT) is a powerful and well-studied approach which has been applied in many scenarios such as validating the confidentiality and integrity of a system [8]–[10]. In this approach, sensitive data flow is explored through tracking information flow from sensitive/taint sources to sensitive/taint targets.

A. Information Flow Policy

In IFT systems, information flow policies are defined to configure the taint source, taint target and tracking rules for taint propagation. These definitions are presented in the following sections. Specific tracking rules are designed to address the idiosyncrasies of analog/mixed-signal circuits.

Taint Sources. The taint source is set up in two steps based on the abstracted model of charge-domain Trojans. Firstly, capacitors that are larger than a threshold are recognized to localize possible taint sources. The reason is that to perform an attack or explore a vulnerability, there must be special properties in charge-domain Trojans. The charge-sharing Trojans have to include a capacitor whose capacitance is larger than a threshold calculated depending on Equation (6). The frequency f_{SW} in the equation is decided by the clock speed of the circuits-under-testing. For capacitive-coupling Trojans, besides large parasitic capacitors between two wordlines, the charge leakage of each cell is also taken into consideration.

In the second step, we search around the special structure to match the abstracted model of charge-domain leakage paths. If a match is found, then we define the matched architecture as a taint source. The taint is propagated from the ports of the architecture, which are used to activate leakage paths. For instance, if the architecture in Figure 2(i) is recognized as a taint source, the propagation is tracked from ports A and B.

Taint Targets. To activate the charge-domain Trojans, adversaries must employ a user controllable portion in the circuit, which can drive signal to reach the leakage paths. We define a signal that invokes the leakage paths successfully as an effective signal. Thus, we define the user controllable portion as the taint targets. As an example, charge-sharing leakage paths are employed through a toggling signal from a flip-flop. Therefore, all user-accessible flip-flops in charge-sharing attacks are identified as targets.

Taint Propagation. After being produced from the taint target, the effective signal may flow through analog/mixed-signal circuits, which is a combination of components like MOSFETs, capacitors, diodes, resistors, etc. We name this flow path as effective signal flow. Then the taint propagation is defined to follow the opposite direction of the effective signal flow. For the components like diodes and resistors, the effective signal can pass them directly. The direction of effective signal is the same as current, and the taint propagation direction is backward. The terminals like V_{DD} and V_{SS} are independent of the flow of the effective signal, hence they are the termination of the effective signal flow and taint propagation. At this point, we assume there are combinational gates, which are composed of MOSFETs, between the taint source and taint target. Similarly as above, among connected MOSFETs, the taint propagates from the Gate of one MOSFET to the Drain or the Source of the other MOSFET. Inside a MOSFET, the taint is transmitted from the Drain or the Source to the Gate. We show examples about the taint propagation inside MOSFET, Diode, and Resistor in Figure 3.

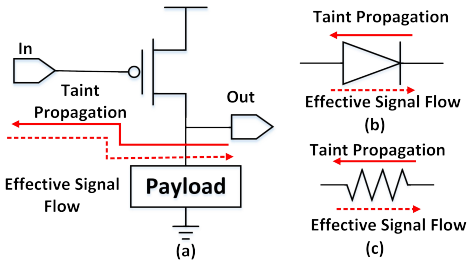

Figure 3: Taint propagation in MOSFET, Diode, and Resistor.

B. Taint Tracking Tool

An automatic taint tracking tool is developed to realize the information flow policy using Python. We implement the tool that formalizes a data flow graph (DFG) from a transistor layout of the circuit. Each transistor in the circuit maps to a node in the DFG. I/O ports of the transistor are designed as parameters of the node. Connections among transistors are identified as edges among nodes.

Depending on the abstracted models of charge-domain information leakage paths, the tracking tool performs analysis on DFG and recognizes the taint sources. The tool then tracks the taint by searching in the opposite direction of the effective signal's flow. Any endpoints like V_{DD} and V_{SS} are treated as the termination of the current taint propagation. Once the taint reaches targets, a "Trojan detected" warning is raised.

V. Experimental Results

In the experiment, we show the insertions of charge-sharing Trojans to an experimental RISC microcontroller and demonstrate the corresponding detection using our taint tracking tool.

A. Charge Sharing Trojans Insertion

For the benchmark, we implement a microcontroller based on the standard GF/IBM 130nm process and embed several Trojans after place & route from the attackers' view. The microcontroller is an experimental Reduced Instruction Set Computer (RISC) processor with an 8K address space [11]. Specifically, the microcontroller is first obtained at the functional level and written in Verilog. We then synthesize the RTL design and perform place&route to get the layout. In the original version of benchmark, there are 317 combinational cells and 241 sequential cells in total. Based on the layout, several charge-sharing Trojans are inserted into the empty space which is occupied by fillers. After Trojan insertions, the total number of extracted transistors are 4373 in the extracted netlist, which includes 2305 NFETs and 2068 PFETs.

In this demonstration, the Trojans are triggered by program controllable registers such as Instruction Registers, Address Registers, Arithmetic Logic Unit Output Registers, and Program Counter Registers. Meanwhile, to decrease the visibility, the leakage paths are connected with the outputs of registers via additional logic instead of being directly connected. On the other side of the leakage paths, some important state registers are connected as payloads, such as the state registers of the CPU state machine control module and the state registers of the multi-clock generator. Therefore, through manipulating the charge sharing leakage paths, the adversaries are able to fundamentally affect the functions of the CPU and even make the whole chip temporarily paralyzed.

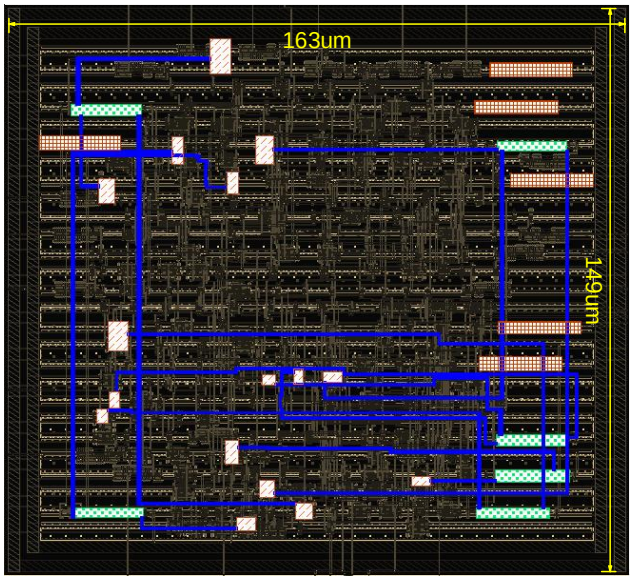

Figure 4: The layout of microcontroller embedded with several charge-sharing Trojans variants. All Trojans (marked as green) are automatically detected and are distinguished from suspicious capacitors with no threat (marked as red lattice shadow). The threatened signal paths (marked as blue) and registers (marked as red slash shadow) are identified using the proposed IFT policy.

In Figure 4, we show the layout of the benchmark after the Trojans insertion. We insert 6 charge-sharing Trojans including 2 (f), 2 (g), 1 (i) and 1 (j) of Trojans presented in Figure 2. Additionally, 6 benign normal large capacitors are marked as a red lattice shadow, which are used to test the false positive of the proposed detection.

B. Taint Tracking Tool Application

After obtaining the layout from the untrusted back-end house, we extract the circuit information from the layout via back-end verification tools and analyze the extracted SPICE netlist based on the proposed IFT method from the defenders' perspective. In detail, at first, our taint tracking tool formalizes the SPICE files to the data flow graphic (DFG), and then localizes all the capacitors larger than a certain threshold.

We calculate the threshold by direct characterization of the process. For the microcontroller implemented using the 130nm process, the leakage current I_g of MOS capacitors are small within several $pA/\mu m^2$. So the charge sharing leakage paths are easily triggered. In order to avoid activating the leakage paths by accident, the capacitor C_1 must be large enough. So, we set the threshold size of suspicious MOS capacitors at $W \times L \leq 1.5\mu m^2$ and the corresponding minimum trigger frequency is at $200MHz$, which is program controllable but higher than the frequency of normal operations. It is safe to say any capacitors larger than the threshold size are suspicious without false negative and need to be further analyzed. If the microcontroller is implemented in more advanced technology nodes such as the 40nm process, the leakage current I_g is much larger, thus the sampling capacitors C_2 must be large enough to make the minimum trigger frequency achievable, as described in Equation (6). We still can set a safe threshold size of MOS capacitors based on process data and find suspicious capacitors efficiently.

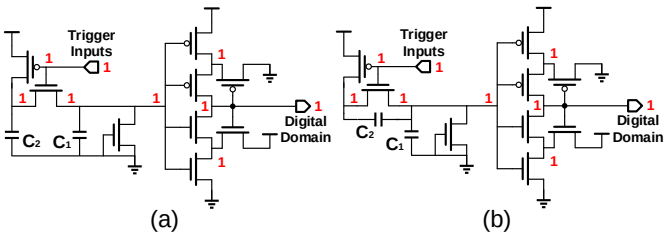

Figure 5: Security analysis of VeriCoq. (a) a correctly detected A2 Trojan; (b) a false positive detection of a benign switched-capacitor circuit that is invalid to carry out A2 attack.

The tool then searches for charge-sharing leakage paths' structures around the capacitors. If the taint source exists, the taint propagation is tracked in the opposite direction of effective signal flow. The registers in the microcontroller are identified as taint targets. Once any propagation reaches the taint target, the charge-sharing leakage paths are detected.

C. Results and Analysis

Our experimental results show that all the embedded charge-sharing Trojans have been detected, and none of the normal working capacitors are detected as Trojan. It means that our approach can expose Trojans effectively and precisely without false positive. In Figure 4, the propagation paths between the user-accessible registers and charge-sharing Trojans are highlighted in blue. Also, the validation of information policies in the given benchmark only takes 5.8 seconds.

VI. Comparisons to Existing Solutions

A. IFT Solution in Mixed-Signal Domain: VeriCoq

VeriCoq presents an IFT based solution which tracks sensitive signals in the mixed-signal domain. An A2 leakage path is detected by VeriCoq as shown in Figure 5(a), where the integer 1 stands for the sensitive label [5]. When the trigger inputs are set to be sensitive, the labels can be propagated to the digital domain via an A2 leakage path. However, such tracking method leads to a high false positive rate with many benign circuit structures be mis-classified as information leakage paths. A counter example is shown in Figure 5(b). We show an implementation of the switched-capacitor topology in Figure 2(c), which is equivalent to open circuit between trigger inputs and the digital domain, as discussed in Section III-D. Using VeriCoq, the sensitive label will still be propagated from trigger inputs to the digital domain as well. In the proposed detection method, the utilization of abstracted models for charge-domain information leakage paths helps localize information leakage paths as a taint source accurately, which prevents the false positive from happening. Additionally, as the propagation direction is opposite to the effective signal flow, the propagation will be terminated when it reaches the Gates of MOSFETs connected with trigger inputs.

B. Runtime Detection: R2D2

An on-chip runtime A2 detection method is implemented in prior work [4]. The method inserts monitor circuits to check the frequency of toggling on several signals which have the potential to activate the A2 Trojan. An interruption will be given if the toggling frequency of a register is higher than a threshold. As a runtime monitoring method, R2D2 suffers from area and power overheads. Our approach, as a static analysis, avoids such issues. In addition, false positive will be produced by using R2D2 because of various applications running in CPU, while our method is less prone to false positive as mentioned in the above. Overall, R2D2 only considers the registers' toggling frequency, which is an excessively specific behavior of A2. Thus the R2D2 is developed particularly as an A2 countermeasure. The abstracted models of charge-domain Trojans are more general in our solution, hence we can perform broader analog Trojans detection.

VII. Conclusion

In this paper, charge-domain Trojans, a broadly-defined model abstraction for capacitor-based vulnerabilities, are proposed to expose the stealthy yet harmful threats in analog/mixed-signal domain. Using features extracted from the charge-domain Trojans models, an IFT based solution is presented to detect charge-domain leakage paths, and a taint tracking tool is developed to realize the solution. In our future work, we will apply the developed method to large-scale circuits using more advanced technology nodes.

Acknowledgement

This work was partially supported by Army Research Office (W911NF-17-1-0477), Cisco and the National Science Foundation (CNS-1657562).

References

[1] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, "Trojan detection using ic fingerprinting," in *Security and Privacy, 2007. SP'07. IEEE Symposium on*. IEEE, 2007, pp. 296–310.

[2] K. Yang, M. Hicks, Q. Dong, T. Austin, and D. Sylvester, "A2: Analog malicious hardware," in *Security and Privacy (SP), 2016 IEEE Symposium on*. IEEE, 2016, pp. 18–37.

[3] Y. Kim, R. Daly, J. Kim, C. Fallin, J. H. Lee, D. Lee, C. Wilkerson, K. Lai, and O. Mutlu, "Flipping bits in memory without accessing them: An experimental study of dram disturbance errors," in *ACM SIGARCH Computer Architecture News*, vol. 42, no. 3. IEEE Press, 2014, pp. 361–372.

[4] Y. Hou, H. He, K. Shamsi, Y. Jin, D. Wu, and H. Wu, "R2d2: Runtime reassurance and detection of a2 trojan," in *2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 2018, pp. 195–200.

[5] M.-M. Bidmeshki, A. Antonopoulos, and Y. Makris, "Information flow tracking in analog/mixed-signal designs through proof-carrying hardware ip," in *Proceedings of the Conference on Design, Automation & Test in Europe*. European Design and Automation Association, 2017, pp. 1707–1712.

[6] Y. Shiyanovskii, F. Wolff, A. Rajendran, C. Papachristou, D. Weyer, and W. Clay, "Process reliability based trojans through nbti and hci effects," in *Adaptive Hardware and Systems (AHS), 2010 NASA/ESA Conference on*. IEEE, 2010, pp. 215–222.

[7] D.-H. Kim, P. J. Nair, and M. K. Qureshi, "Architectural support for mitigating row hammering in dram memories," *IEEE Computer Architecture Letters*, vol. 14, no. 1, pp. 9–12, 2015.

[8] A. C. Myers and B. Liskov, "A decentralized model for information flow control," in *Proceedings of ACM Symposium on Operating Systems Principles (SOSP)*, 1997, pp. 129–142.

[9] D. Zhang, Y. Wang, G. E. Suh, and A. C. Myers, "A hardware design language for timing-sensitive information-flow security," *ACM SIGPLAN Notices*, vol. 50, no. 4, pp. 503–516, 2015.

[10] X. Li, V. Kashyap, J. K. Oberg, M. Tiwari, V. R. Rajarathinam, R. Kastner, T. Sherwood, B. Hardekopf, and F. T. Chong, "Sapper: A language for hardware-level security policy enforcement," in *ACM SIGARCH Computer Architecture News*, vol. 42, no. 1, 2014, pp. 97–112.

[11] "Simplified risc cpu design code," https://download.csdn.net/download/a14730497/4445255, accessed September 15, 2018.