Run-Time Hardware Trojan Detection in Analog and Mixed-Signal ICs

Antonios Pavlidis, Eric Faehn, Marie-Minerve Louërat, Haralampos-G. Stratigopoulos

Run-Time Hardware Trojan Detection in Analog and Mixed-Signal ICs

Antonios Pavlidis*, Eric Faehn†, Marie-Minerve Louërat*, Haralampos-G. Stratigopoulos*
*Sorbonne Université, CNRS, LIP6, Paris, France
†ST Microelectronics, Crolles, France

***Abstract*—Hardware Trojan (HT) insertion is a major security threat for electronic components that demand a high trust level. Several HT attack mechanisms have been demonstrated to date, and several HT prevention and detection countermeasures have been proposed to thwart HT attacks. Given the multitude of HT attack mechanisms, run-time monitors for HT detection are used as a last line of defense. In this paper, we propose a run-time monitoring methodology for HT attack mechanisms affecting the analog and mixed-signal (AMS) sections of an Integrated Circuit (IC). The methodology is based on the Symmetry-based Built-In Self-Test (*SymBIST*) principle that relies on distributing invariances across the IC and continuously checking for their compliance. Detection of various HT attacks are demonstrated on a Successive Approximation Register (SAR) Analog-to-Digital Converter (ADC) IP at transistor-level.**

I. Introduction

The globalization of the Integrated Circuit (IC) supply chain where design and manufacturing tasks are outsourced to third parties has given rise to several hardware security and trust threat scenarios. Among them is Hardware Trojan (HT) attacks that consist of performing a malicious modification in the hardware which when triggered it executes an offensive payload, such as denial-of-service, performance degradation, or creation of a covert channel for leaking sensitive information off-chip, e.g., cipher keys [1]–[3].

There is a large body of literature proposing HT attack mechanisms. The majority of them targets digital portions of an IC, but several of them have been proposed specifically for the analog and mixed-signal (AMS) portions of an IC. A concise review of HT attack mechanisms is provided in Section II. In parallel, HT prevention and detection countermeasures are being proposed that can be generic or target specific HT attack mechanisms. A concise review of HT prevention and detection countermeasures is provided in Section III. From the attacker's perspective, the goal is to insert a HT attack mechanism that is stealthy with small footprint, such that it is capable of evading known HT prevention and detection countermeasures. Given the multitude of HT attack mechanisms, the defender is forced to combine many of these countermeasures, thus increasing costs. Still, there is no guarantee that every known HT attack mechanism is addressed or that undocumented and new HT attack mechanisms will be prevented or detected. Therefore, as a last line of defense run-time monitors need to be deployed into the design aiming at detecting HT activation in real-time with low-latency, thereafter switching the chip to a safe mode operation. A concise review on run-time monitors for HT detection is provided in Section IV. The existing run-time monitors have been demonstrated for digital ICs and/or are specific to a HT attack mechanism.

In this paper, we propose a generic run-time HT monitoring methodology for AMS portions of an IC. Naturally, we can think of the several monitors that have been proposed for AMS ICs and are part of Design-for-Test (DfT) or Built-in Self-Test (BIST) infrastructures targeting hardware-level fault detection. This is because both a hardware-level fault and a HT payload result in internal signal or topology modification albeit the root-cause being different. Furthermore, if we can rely on reusing existing DfT or BIST infrastructure, there will be no extra required on-chip resources for run-time HT detection. A BIST that runs on-line as part of a functional safety loop in critical applications naturally can have an auxiliary use for run-time HT detection. The problem is that designing on-line BIST for AMS ICs is challenging and few solutions exist today.

More specifically, existing monitors targeting hardware-level fault detection in AMS ICs include amplitude detectors [4], [5], current sensors [6], [7], jitter estimators [8], [9], temperature sensors [10], and Pseudo-Random Bit Sequence (PRBS)-based sensors [11]. However, none of these monitors are appropriate for run-time HT detection. The reason is that they do not run on-line with the operation of the circuit requiring a specific test stimulus and/or post-processing of test responses. On-die process control monitors used for predicting performance variations due to process instabilities [12], [13] are not appropriate either since they are not electrically connected to the circuit, thus they are not actuated by the HT. On-line monitoring techniques for AMS ICs exist only for fully-differential [14]–[16] and linear time-invariant [17]–[19] blocks. However, the objective is to come up with a generic solution applicable to any AMS IC.

To this end, we rely on a recently proposed generic on-line BIST solution for AMS ICs called Symmetry-based BIST (*SymBIST*) [20]. The working principle is based on distributed monitors (or checkers) that construct invariances and check their compliance. An invariance is a signal that by construction in error-free operation stays within a tolerance window regardless the input to the circuit. When an error occurs, one or more invariances are violated and the corresponding monitors flag an alert signal. The same *SymBIST* infrastructure is used for post-manufacturing defect-oriented test with high

This work was supported by the ANR STEALTH project under Grant N° ANR-17-CE24-0022-01 and the ANR EDITSoC project under Grant N° ANR-17-CE24-0014-02.

978-1-6654-1060-1/22/$31.00 ©2022 IEEE

fault coverage and for concurrent error detection targeting single event upsets, latent defects, and aging [20]. *SymBIST* has also been demonstrated for fault diagnosis resulting in high diagnosis resolution and fast diagnosis cycle [21].

In this paper, we demonstrate that the *SymBIST* principle can be reused for run-time detection of HT payload activity into AMS ICs. The HT payload activity breaks the compliance of one or more invariances and can be detected in real-time regardless the running input. We demonstrate this operation on an industrial Successive Approximation Register (SAR) Analog-to-Digital Converter (ADC) at transistor-level for different HT attack mechanisms. Since invariances cover the entire design, in principle any HT payload activity can be detected, thus the method is agnostic to the HT attack mechanism. The demonstration is based on digital-to-analog HTs [22], [23] and HT payloads in the form of bit line flips in the digital section of the SAR ADC.

The rest of the article is organized as follows. In Section II, we provide a review of HT attack mechanisms in the digital and AMS domains. In Section III, we discuss existing HT prevention and detection countermeasures. In Section IV, we discuss in more detail run-time HT detection. In Section V, we present the *SymBIST* working principle for run-time HT detection. In Section VI, we present the SAR ADC IP architecture with the embedded *SymBIST* infrastructure. In Section VIII, we demonstrate run-time HT detection using *SymBIST*. Section IX concludes this article.

II. HT ATTACK MECHANISMS

A. Digital domain

Classical HT attack mechanisms are the combinational and sequential [2]. A combinational HT monitors a set of nodes and is triggered on the simultaneous occurrence of rare node conditions, subsequently delivering its payload to another node by flipping the node value. A sequential HT is triggered instead with a sequence of conditions. There is a multitude of more sophisticated HT attack mechanisms, including silicon wear-out mechanisms [24], hidden side-channels [25], changing dopant polarity in active areas of transistors [26], siphoning charge from victim wires known as the A2 attack [27], [28], activating a row in DRAM to corrupt data in nearby rows known as the rowhammer attack [29], and scan attacks [30], e.g., theft of secret keys [31], [32], tampering Intellectual Property (IP) blocks [33], and memory damping [34]. Several benchmarks can be found in Trust-Hub [35].

B. AMS domain

HT attack mechanisms for AMS portions of an IC include bringing the circuit into an undesired state or operation mode [36]–[40], digital-to-analog HTs [22], [23], and HTs in wireless ICs [41]–[47]. Digital-to-analog HTs apply to a System-on-Chip (SoC) comprising digital and AMS IP blocks. Their trigger mechanism is hidden inside a digital IP and the payload is transferred via the common test infrastructure to the victim AMS IP. The payload consists in setting the AMS IP to a partial test mode or altering the configuration of the AMS IC to set it to an undocumented operation mode. HTs in wireless ICs aim at leaking sensitive information from the transmitter within a legitimate signal transmission. A rogue receiver can listen to the transmission to recover the sensitive information, while the legitimate receiver is inconspicuous and does not realize the information leaking.

III. HT COUNTERMEASURES

Several prevention and detection techniques are available to defenders to thwart HT attacks. Their applicability depends on the HT insertion phase and level.

If the attack is staged by an Electronic Design Automation (EDA) tool provider or by a third-party IP (3PIP) provider, then the design owner can use the following detection techniques: (a) functional verification of the 3PIP cores [48]; (b) structural analysis of Hardware Description Language (HDL) codes [48]; (c) generation of test patterns to expose the HT [49]; (d) specific simulation benches to magnify the effect of the HT, i.e., performing aging simulations along with over-clocking [50]; (e) search methods for unused components during design-time verification, which thereafter can be removed as potentially suspicious [51]; and (f) Information Flow Tracking (IFT) methods that track the propagation of sensitive data and verify that they do not reach unauthorized sites in the design [28], [52]–[55].

If the attack is staged by the foundry, pre-silicon prevention methods include: (a) filling in with functional filler cells all unused spaces on the layout, which are most likely insertion areas for the HT, and checking if those have changed [56]; and (b) design obfuscation, for example using locking [57], [58], camouflaging [59], [60], or split manufacturing [61], [62], aiming at obscuring the circuit functionality so as to make it difficult for the attacker to insert the HT.

Post-silicon HT detection methods include: (a) testing [49]; (b) reverse-engineering [63]; (c) optical circuit analysis aiming at measuring optical emissions of the IC and comparing them with a trusted emission image of a "golden" IC [64]; (d) statistical side-channel fingerprinting (SSCF) aiming at exposing the HT by its effect on parametric measurements, i.e., delay, power, temperature, etc. [65], [66]; and (e) using on-chip monitors for run-time HT detection [67]–[76].

There exist also defenses aiming at improving the trust in the on-chip test infrastructure [30], including access authentication [77], [78], assuring data confidentiality and integrity [33], and on-line detection of test pattern compliance [79].

These HT prevention and detection countermeasures were developed for digital portions of an IC, but they are generic and can be applied to AMS portions of an IC as well, or the principle of operation can be adapted to AMS ICs. For example, SSCF is a generic countermeasure that has shown to be very effective for AMS ICs since several parametric measurements can be defined on AMS ICs on which a HT inevitably leaves traces [45]. Design obfuscation techniques have also been recently proposed for AMS ICs, including locking [80]–[84] and camouflaging [85], [86]. IFT is adapted for AMS ICs in [87].

IV. Run-time monitors for HT detection

Several run-time monitors for HT detection have been proposed in the literature [67]–[76]. In [67], it is assumed that the operating system is trusted and it is modified to create run-time checks on the hardware. In [68]–[70], monitors are proposed that detect current and power HT traces on the power grid. In [71], [72], temperature sensors are exploited to detect deviations in power/thermal profiles caused by HT activation. In [73], a run-time detection of the HT attack in wireless ICs described in [44], [45], [47] is proposed. The HT attack consists in leaking data into minute modifications in the parameters of the transmitted signal, such as amplitude and frequency. Based on the observations that the circuit draws specific current for a '1' or '0' signal transmission and that the HT manipulating the transmitted signal has a direct impact on the current drawn, the power supply is continuously monitored checking for equality of drawn current across consecutive transmissions of a given number of '1s' and '0s'. This expected equality is an invariant side-channel fingerprint extracted concurrently with the normal operation of the wireless IC and its compliance is evaluated through a trained on-chip classifier. In [74], it is proposed to identify high-level and critical behavioral invariants of a processor inspired from assertion-based verification, and a Hardware Property Checker (HPC) is designed that verifies these invariants at run-time. Another approach uses temporal logic assertions to automatically synthesize monitor circuits [75]. A run-time detection of the A2 Trojan [27], [28] is proposed in [76] that works by guarding a set of concerned victim signals and initiating a hardware interrupt request when abnormal toggling events occur.

V. *SymBIST* for run-time HT detection

The first step in a *SymBIST* implementation is to identify or construct invariances. An invariance is an ideally constant signal generated by monitoring internal nodes of the circuit. Formally, let x_i denote a voltage or current signal in node i. An invariance j built from signals in nodes $1, \cdots, n$ is an equation in the form $f_j(x_1, \cdots, x_n) = 0$. The equality holds for any input to the circuit.

Identifying or constructing invariances is a circuit-specific problem, but several invariances are generic and are found in any AMS IC. Examples include the summation of two fully-differential or complementary signals and the subtraction of two identical signals from replicated sub-blocks.

The goal of the designer is to create several invariances covering the entire design, in the sense that if an error occurs, in our case induced by a HT activation, it is manifested by the violation of at least one invariance. In practice, the number of invariances can be small thanks to feedback loops existing inside an AMS IC. For example, the SAR ADC case study circuit in Section VI is fully covered by 7 invariances in total.

An invariance is constructed and continuously monitored by a checker, as shown in Fig. 1. The checkers' inputs are the nodes' signals and their output is a 1/0 bit denoting compliance/violation of the invariance. The checkers' outputs

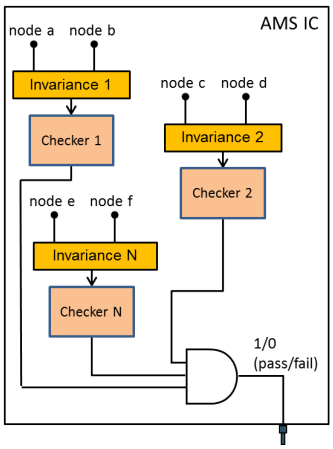

Fig. 1. *SymBIST* principle of operation.

are driven to an AND gate to generate a single 1-bit pass/fail decision. Since many invariances have the same form, i.e., $x_1+x_2 = \alpha$ or $x_1-x_2 = \beta$, the checker design can be reused. To avoid loading the circuit and inducing a performance penalty, checkers are designed with a high input impedance.

In practice, a true invariance does not exist due to noise and process, voltage, and temperature (PVT) variations. Thus, a checker implements a tolerance window $[-\delta, \delta]$, $\delta > 0$, and monitors whether the invariance stays within the tolerance window, i.e., $|f_j(x_1, \cdots, x_n)| < \delta$. The size of the tolerance window is set to avoid false positives and the choice is made by performing corner and Monte Carlo simulations.

The underlying idea is that any error in the circuit operation will break one or more of the invariant properties distributed across the circuit. *SymBIST* can be used for post-manufacturing testing and fault diagnosis, where specific test stimuli and test benches can be used towards high fault coverage and fault diagnosis resolution. A fault is deemed detected by a test stimulus if it violates at least one invariant property [20]. For fault diagnosis, the diagnostic measurement pattern is a bit-string vector $1\times(N\times k)$, where N is the number of checkers and k is the number of clock cycles. The goal here is to craft test stimuli and select test benches such that any two faults have distinguishing diagnostic measurement patterns with a Hamming distance of minimum one [21]. *SymBIST* can also be reused concurrently with the operation for on-line test since invariant properties are input-independent [20]. This is exactly the *SymBIST* usage exploited for run-time HT detection in this work. Thus, assuming that *SymBIST* is already in place and is used for concurrent on-line testing in the context of a mission-critical application, it is reused for run-time HT detection with no extra cost. It may be needed, however, to define new invariances targeting specifically known HT attack mechanisms. For the SAR ADC case study circuit in Section VI, a new invariance is defined to call attention to a particular HT attack mechanism. This new invariance can further improve fault coverage, fault diagnosis resolution, and coverage of run-time errors in general regardless the root-cause, i.e., single event upsets, latent defects, or aging.

If the checkers are permanently invalidated by the attacker,

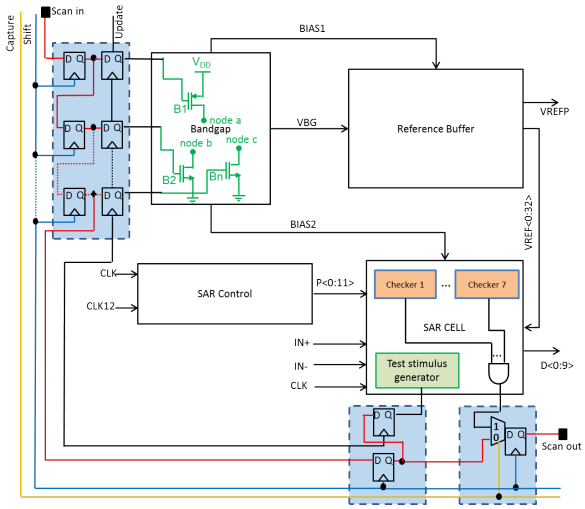

Fig. 2. SAR ADC top-level architecture.

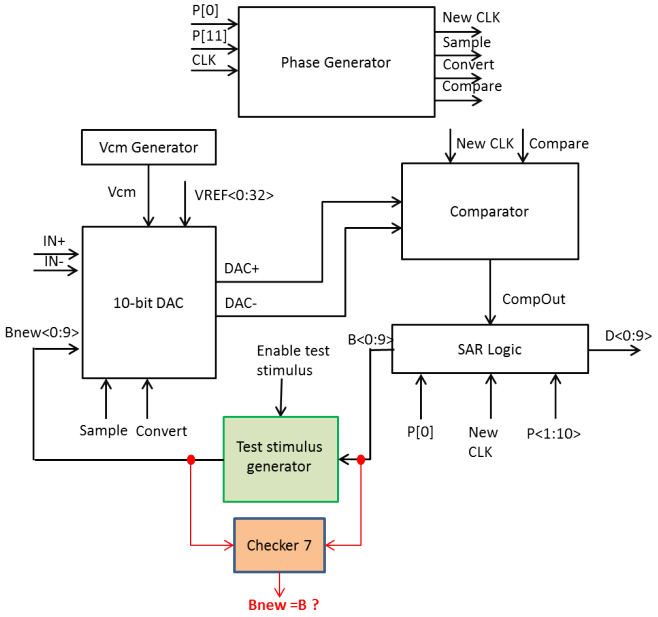

Fig. 3. SAR CELL architecture.

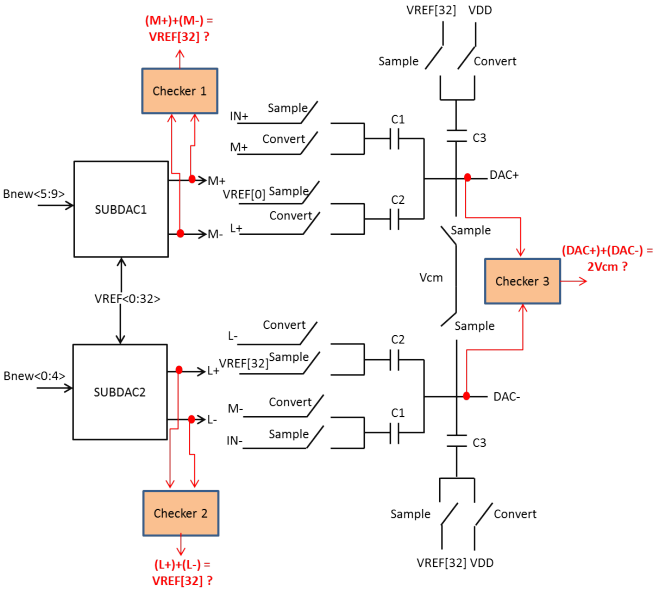

Fig. 4. 10-bit DAC architecture.

then this is straightforwardly detectable at test time since checkers are equipped with a self-test mechanism and are tested prior to their usage [20]. It is assumed that the attacker cannot compromise the checkers simultaneously at the moment of HT activation since this would require significant routing to reach all distributed checkers and, thereby, the HT attack mechanism stops having small footprint and is easily recognizable by layout inspection or reverse-engineering.

VI. Case Study

Our case study is a 10-bit SAR ADC IP by STMicroelectronics designed in 65nm CMOS. The top-level architecture is shown in Fig. 2 along with the embedded *SymBIST* infrastructure and the test access and control mechanism. The top-level blocks are the SAR CELL that implements the main data conversion algorithm, the digital SAR Control, the bandgap that generates the different analog biases, and the reference buffer that generates the different reference voltages used by the data conversion algorithm. All the checkers, 7 in total, are implemented inside the SAR CELL. This is because any errors in the analog biases, reference voltages or digital SAR Control will be reflected in the SAR CELL operation. A test stimulus generator is inserted inside the SAR CELL. Moreover, topology modifications are implemented inside the critical bandgap [88]. Topology modifications are enabled by distributed digitally-controlled Pull-Up (PU) and Pull-Down (PD) transistors that connect internal nodes to power supply and ground, respectively. The underlying principle is that by topology modification defects will be better exposed and invariance violation will be more pronounced. The test access and control mechanism is proposed in [89] and is compatible with the IEEE Std. 1687. The digital control of the PU/PD transistors, the enable bit of the test stimulus generator, and the 1-bit output of *SymBIST* are accessed by the scan chain traversing the different IPs of the SoC. Fig. 2 is simplified not showing other IPs appended to the scan chain.

Figs. 3-7 show a top-down breakdown of the SAR ADC architecture indicating the invariances and their positioning inside it. Fig. 6 shows the top-level architecture of the bandgap. It has a total of 13 PU/PD transistors embedded, with only 3 shown at the top-level as the rest are inserted into the other sub-blocks, i.e., start-up circuit, self-biased operational transconductance amplifiers (SOTAs), and output stage. The test stimulus generator shown in Fig. 7 is used only for off-line defect-oriented test and diagnosis. It generates all bit combinations at the inputs of the two 5-bit SUBDACs inside the 10-bit DAC of the SAR CELL. It is based on a 5-bit counter followed by a shuffler to implement a non-incremental counting as this exercises more intensively the SAR ADC. The same test stimulus is used for testing simultaneously all invariances. The test duration is very short and equals $2^5 \cdot (1/f_{clk}) = 0.206\mu s$, where f_{clk} is the clock frequency. The test stimulus generator is enabled by one bit that controls

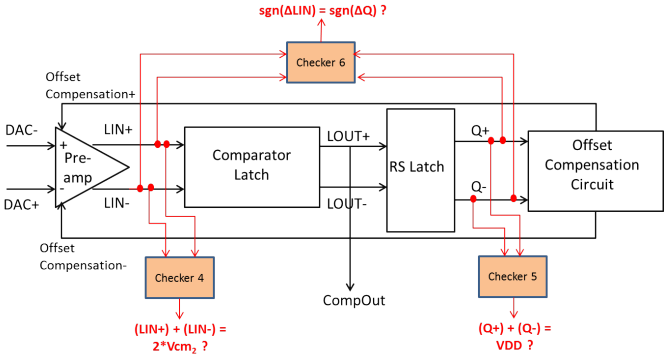

Fig. 5. Comparator architecture.

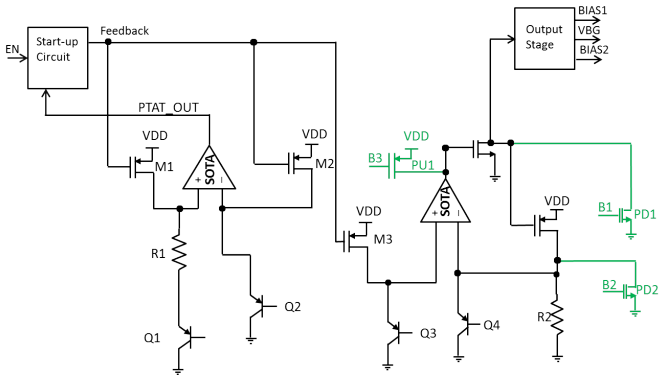

Fig. 6. Bandgap architecture.

a number of multiplexers. In normal mode, the digital bits $B < 0,9 >$ during the conversion are fed to the input of the 10-bit DAC. At the end of the conversion, the ADC output is $D < 0,9 >= B < 0,9 >$. In test mode, the test stimulus generator is connected to the input of the 10-bit DAC, disconnecting the SAR Logic block from the 10-bit DAC.

In total, 7 invariances are built and are continuously checked. All invariances are positioned inside the SAR CELL block, as shown in Fig. 2, and can reflect errors in any block of the SAR ADC architecture. Invariances 1-3 are positioned inside the DAC of the SAR CELL, as shown in Fig. 4, invariances 4-6 are positioned inside the comparator of the SAR CELL, as shown in Fig. 5, and invariance 7 is positioned at the top-level of the SAR CELL, as shown in Fig. 3. Invariances 1-2 and 5 are built on complementary signals whose sum should equal a specific value. Invariances 3-4 are built on fully-differential signals which have equal magnitude but opposite polarity and, thereby, their sum should equal twice the common mode voltage. The checkers for invariances 1-5 are simple analog comparators implementing a window comparison [20]. Invariances 6-7 are built on digital signals and their checkers are implemented with digital gates. Invariance 7 is not part of the original *SymBIST* infrastructure for this SAR ADC IP [20], [21] and has been added to detect a specific HT attack, as it will be described in Section VII.

The *SymBIST* infrastructure has an area overhead of less than 5%. Also the power consumption for the on-line test use is negligible. Finally, it has been verified that there is no performance penalty to the SAR ADC IP due to the insertion of the test stimulus generator, PU/PD transistors in the bandgap, and checkers [20], [21].

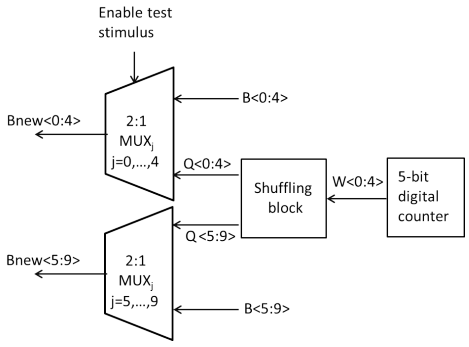

Fig. 7. Test stimulus generator.

VII. HT ATTACK SCENARIOS

We consider that the SAR ADC operates in normal mode with the invariances being continuously checked. We consider two known HT payload mechanisms, namely digital-to-analog HT attacks [22], [23] and bit line flips inside digital sections of the SAR ADC. In digital-to-analog HT attacks, the HT is triggered inside a dense digital IP of the SoC and transfers its payload to the BIST interface of the SAR ADC with the test access and control mechanism. In particular, the payload consists in enabling a random combination of PU/PD transistors inside the bandgap and/or the test stimulus generator inside the SAR CELL. Thus, during normal mode, upon HT triggering, the SAR ADC is set to a possibly undocumented test mode and its normal operation is interrupted. We consider only the HT payload as such a digital-to-analog HT attack can make use of any HT trigger mechanism, i.e., combinational, sequential, A2 Trojan, etc. Bit line flips is a commonly used HT payload and, again, several HT trigger mechanisms can be employed. In short, we do not make any assumption on the HT trigger and we focus only on the HT payload.

A transient transistor-level simulation of the SAR ADC is very time-consuming, in the order of 1 hour to record the output for a minimum time interval where the HT activation and immediate detection occur. Thus, it is not feasible to exhaustively simulate all the above HT payloads which are in the order of thousands. We performed several simulations with random selection of HT payloads and in every case the HT payload was detected with low latency. In Section VIII, we illustrate the operation for 3 different HT payloads.

VIII. RESULTS

Without loss of generality, we use a piece-wise running input to the SAR ADC. The HT is triggered during normal operation. We demonstrate run-time HT detection for 3 different HT payload cases, namely:

- Case 1: Enabling a PU transistor inside the bandgap.
- Case 2: Enabling the test stimulus generator inside the SAR CELL.
- Case 3: Bit line flip inside SUBDAC1 of the 10-bit DAC.

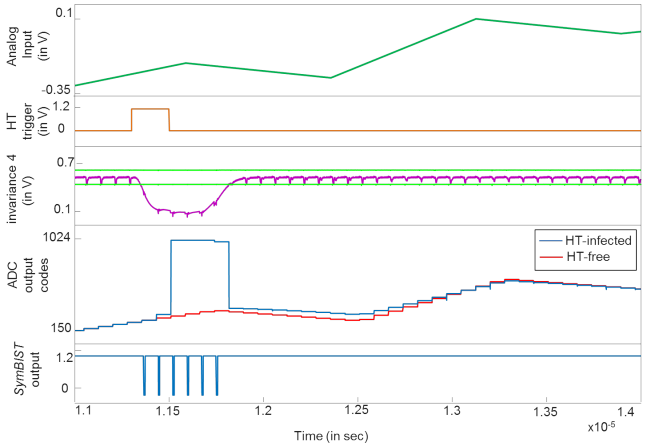

Fig. 8. Run-time detection of HT enabling a PU transistor inside the bandgap.

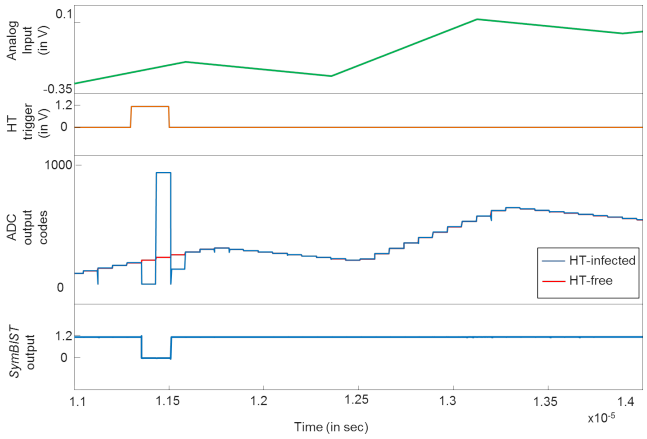

Fig. 9. Run-time detection of HT enabling the test stimulus generator.

Fig. 8 shows the simulation of case 1. The subplots show various signals of interest, i.e., ADC input, HT trigger, invariance and corresponding tolerance window, HT-free and HT-infected ADC output, and *SymBIST* output. The HT is triggered at 11.3 μs. The ADC output starts diverging from the HT-free operation a few tens of ns later and very quickly it saturates to the largest code 1024. Note that before the HT activation the HT-free and HT-infected outputs are identical, showing that the monitor insertion is non-intrusive resulting in no ADC performance penalty. Similarly, once the HT trigger is off, the ADC output converges quickly to the HT-free operation. The HT triggering results in violation of most invariances. Fig. 8 illustrates only invariance 4 positioned inside the comparator of the SAR CELL. As it can be seen from Fig. 8, the invariance is within the tolerance window before the HT is triggered and slides outside the tolerance window upon HT activation. As this is an analog invariance, the checker is an analog comparator with a clocked operation to provide a digital output. Thus, during the time the invariance is violated, the checker's output toggles from 1 to 0, as shown from the *SymBIST* global output. The invariance remains violated for a few tens of ns after the HT trigger is off. The asynchronous events, i.e., delay between first violation of invariance and HT trigger, delay between HT trigger and

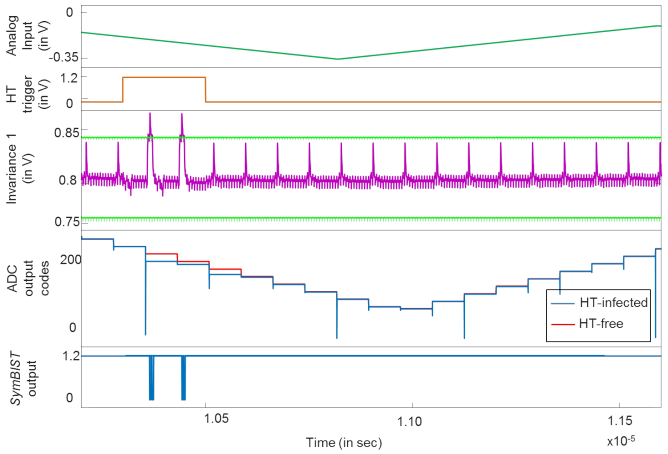

Fig. 10. Run-time detection of HT flipping a bit line inside SUBDAC1 of the 10-bit DAC.

ADC output deviation, and delay for ADC output convergence after the HT is off, are due to the sampling operation and the SAR algorithm itself that successively approximates the input.

Fig. 9 shows the simulation of case 2. Run-time detection of the HT activation is achieved via monitoring of invariance 7. This invariance checks that the SUDACs' input is driven by that SAR Logic block and not by the test stimulus generator during normal operation. As it is a digital invariance, it matches the *SymBIST* output. As it can be seen from Fig. 9, the ADC operation is affected soon after the HT triggering and this is flagged immediately by *SymBIST*.

Fig. 10 shows the simulation of case 3. The HT payload has a more subtle effect on the operation, but still the conversion is erroneous with the output codes being a few tens of codes away from the nominal expected codes. As the bit flip occurs inside SUBDAC1, this HT payload is detected by invariance 1 which monitors the output of SUBDAC1. Due to the subtle effect, the invariance is violated for two short intervals during the duration of the HT activation. The checker's output toggles from 1 to 0 during these two intervals.

IX. Conclusions

There is a growing number and diversity of HT attack trigger and payload mechanisms. HTs can be inserted during different phases of the IC design and could be located anywhere on the chip. Due to this large potential HT attack surface, dealing proactively with HT insertion is important, but does not provide security guarantees. Therefore, as a last line of defense, run-time detection of HT activation is a highly desired feature. In this paper, we demonstrated *SymBIST* for lightweight run-time HT detection in AMS ICs. The idea is to distribute invariances across the design and continuously check for their compliance which holds only in HT-free operation. The *SymBIST* infrastructure is reusable for post-manufacturing testing, fault diagnosis, and on-line concurrent error detection, thus run-time HT detection is an auxiliary usage that comes at no extra cost. Run-time detection of different HT attack scenarios with *SymBIST* was demonstrated on an industrial SAR ADC IP with transistor-level simulations.

REFERENCES

[1] R. Karri, J. Rajendran, K. Rosenfeld, and M. Tehranipoor, "Trustworthy hardware: Identifying and classifying hardware trojans," *Computer*, vol. 43, no. 10, pp. 39–46, Oct. 2010.

[2] S. Bhunia, M. S. Hsiao, M. Banga, and S. Narasimhan, "Hardware trojan attacks: Threat analysis and countermeasures," *Proc. IEEE*, vol. 102, no. 8, pp. 1229–1247, Jul. 2014.

[3] K. Xiao, D. Forte, Y. Jin, R. Karri, S. Bhunia, and M. Tehranipoor, "Hardware Trojans: lessons learned after one decade of research," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 22, no. 1, pp. 6:1–6:23, Dec. 2016.

[4] A. Valdes-Garcia, R. Venkatasubramanian, J. Silva-Martinez, and E. Sanchez-Sinencio, "A broadband CMOS amplitude detector for on-chip RF measurements," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 7, pp. 1470–1477, Jul. 2008.

[5] Y.-C. Huang, H.-H. Hsieh, and L.-H. Lu, "A built-in self-test technique for RF low-noise amplifiers," *IEEE Trans. Microw. Theory Techn.*, vol. 56, no. 2, pp. 1035–1042, May 2008.

[6] M. Cimino, H. Lapuyade, Y. Deval, T. Taris, and J.-B. Bégueret, "Design of a 0.9V 2.45 GHz self-testable and reliability-enhanced CMOS LNA," *IEEE J. Solid-State Circuits*, vol. 43, no. 5, pp. 1187–1194, May 2008.

[7] A. Gopalan, M. Margala, and P. R. Mukund, "A current based self-test methodology for RF front-end circuits," *Microelectronics J.*, vol. 36, no. 12, pp. 1091–1102, Dec. 2005.

[8] S. Sunter and A. Roy, "On-chip digital jitter measurement, from megahertz to gigahertz," *IEEE Design Test Comput.*, vol. 21, no. 4, pp. 314–321, Jul. 2004.

[9] H. Le-Gall, R. Alhakim, M. Valka, S. Mir, H. Stratigopoulos, and E. Simeu, "High frequency jitter estimator for SoCs," in *Proc. 20th IEEE Eur. Test Symp. (ETS)*, May 2015.

[10] L. Abdallah, H.-G. Stratigopoulos, S. Mir, and J. Altet, "Defect-oriented non-intrusive RF test using on-chip temperature sensors," in *Proc. IEEE 31st VLSI Test Symp. (VTS)*, Apr. 2013.

[11] M. Ince *et al.*, "Fault-based built-in self-test and evaluation of phase locked loops," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 26, no. 3, pp. 20:1–20:18, Jan. 2021.

[12] L. Abdallah, H.-G. Stratigopoulos, S. Mir, and C. Kelma, "Experiences with non-intrusive sensors for RF built-in test," in *Proc. IEEE Int. Test Conf. (ITC)*, Nov. 2012, Paper 17.1.

[13] F. Cilici, M. J. Barragan, S. Mir, E. Lauga-Larroze, and S. Bourdel, "Assisted test design for non-intrusive machine learning indirect test of millimeter-wave circuits," in *Proc. 23rd IEEE Eur. Test Symp. (ETS)*, May/Jun. 2018.

[14] N. J. Stessman, B. Vinnakota, and R. Harjani, "System-level design for test of fully differential analog circuits," *IEEE J. Solid-State Circuits*, vol. 31, no. 10, pp. 1526–1534, Dec. 1996.

[15] V. Kolarik, S. Mir, M. Lubaszewski, and B. Courtois, "Analog checkers with absolute and relative tolerances," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 14, no. 5, pp. 607–612, May 1995.

[16] H.-G. D. Stratigopoulos and Y. Makris, "An adaptive checker for the fully differential analog code," *IEEE J. Solid-State Circuits*, vol. 41, no. 6, pp. 1421–1429, Jun. 2006.

[17] A. Chatterjee, "Concurrent error detection and fault-tolerance in linear analog circuits using continuous checksums," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 1, no. 2, pp. 138–150, Jun. 1993.

[18] J. L Huertas, A. Rueda, and D. Vasquez, "Testable switched-capacitor filters," *IEEE J. Solid-State Circuits*, vol. 28, no. 7, pp. 719–724, Jul. 1993.

[19] H.-G. D. Stratigopoulos and Y. Makris, "Concurrent detection of erroneous responses in linear analog circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 5, pp. 878–891, May 2006.

[20] A. Pavlidis, M. M. Louërat, E. Faehn, A. Kumar, and H. G. Stratigopoulos, "SymBIST: Symmetry-based analog and mixed-signal built-in self-test for functional safety," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 6, pp. 2580–2593, Jun. 2021.

[21] A. Pavlidis, E. Faehn, M.-M. Louërat, and H.-G. Stratigopoulos, "BIST-assisted analog fault diagnosis," in *Proc. 26th IEEE Eur. Test Symp. (ETS)*, May 2021.

[22] M. Elshamy, G. Di Natale, A. Pavlidis, M. Louërat, and H.-G. Stratigopoulos, "Hardware Trojan attacks in analog/mixed-signal ICs via the test access mechanism," in *Proc. IEEE Eur. Test Symp. (ETS)*, May 2020.

[23] M. Elshamy *et al.*, "Digital-to-analog hardware Trojan attacks," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 2, pp. 573–586, Feb. 2022.

[24] Y. Shiyanovskii, F. Wolff, A. Rajendran, C. Papachristou, D. Weyer, and W. Clay, "Process reliability based trojans through NBTI and HCI effects," in *NASA/ESA Conf. Adapt. Hardw. Syst.*, Jun. 2010, pp. 215–222.

[25] L. Lin, T. Güneysu M. Kasper, C. Paar, and W. Burleson, *Trojan Side-Channels: Lightweight Hardware Trojans through Side-Channel Engineering*, Berlin, Germany: Springer, 2009.

[26] G. T. Becker, F. Regazzoni, C. Paar, and W. P. Burleson, "Stealthy dopant-level hardware trojans: Extended version," *J. Cryptograph. Eng.*, vol. 4, no. 1, pp. 19–31, Apr. 2014.

[27] K. Yang, M. Hicks, Q. Dong, T. Austin, and D. Sylvester, "A2: analog malicious hardware," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2016, pp. 18–37.

[28] X. Guo, H. Zhu, Y. Jin, and X. Zhang, "When capacitors attack: Formal method driven design and detection of charge-domain trojans," in *Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE)*, Mar. 2019, pp. 1727–1732.

[29] Y. Kim *et al.*, "Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors," in *ACM/IEEE Int. Symp. Comput. Archit.*, Jun. 2014, pp. 361–372.

[30] E. Valea, M. Da Silva, G. Di Natale, M.-L. Flottes, and B. Rouzeyre, "A survey on security threats and countermeasures in IEEE test standards," *IEEE Des. Test*, vol. 36, no. 3, pp. 95–116, Jun. 2019.

[31] B. Yang, K. Wu, and R. Karri, "Secure scan: A design-for-test architecture for crypto chips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 10, pp. 2287–2293, Oct. 2006.

[32] J. Da Rolt, G. Di Natale, M. Flottes, and B. Rouzeyre, "New security threats against chips containing scan chain structures," in *Proc. IEEE Int. Symp. Hardw.-Oriented Secur. Trust*, Jun. 2011.

[33] K. Rosenfeld and R. Karri, "Attacks and defenses for JTAG," *IEEE Des. Test Comput.*, vol. 27, no. 1, pp. 36–47, Jan./Feb. 2010.

[34] Ing. M.F. Breeuwsma, "Forensic imaging of embedded systems using JTAG (boundary-scan)," *Digit. Invest.*, vol. 3, no. 1, pp. 32–42, Mar. 2006.

[35] B. Shakya, T. He, H. Salmani, D. Forte, S. Bhunia, and M. Tehranipoor, "Benchmarking of hardware trojans and maliciously affected circuits," *J. Hardw. Syst. Secur.*, vol. 1, no. 1, pp. 85–102, Mar. 2017.

[36] Z. Liu, Y. Li, Y. Duan, R. L. Geiger, and D. Chen, "Identification and break of positive feedback loops in trojan states vulnerable circuits," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Jun. 2014, pp. 289–292.

[37] X. Cao, Q. Wang, R. L. Geiger, and D. J. Chen, "A hardware trojan embedded in the inverse widlar reference generator," in *Proc. IEEE 58th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2015.

[38] Q. Wang, R. L. Geiger, and D. Chen, "Hardware trojans embedded in the dynamic operation of analog and mixed-signal circuits," in *Proc. Nat. Aerosp. Electron. Conf. (NAECON)*, Jun. 2015, pp. 155–158.

[39] C. Cai and D. Chen, "Performance enhancement induced trojan states in op-amps, their detection and removal," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2015, pp. 3020–3023.

[40] Q. Wang, D. Chen, and R. L. Geiger, "Transparent side channel trigger mechanism on analog circuits with PAAST hardware trojans," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2018.

[41] A. Dutta, D. Saha, D. Grunwald, and D. Sicker, "Secret agent radio: Covert communication through dirty constellations," in *Information Hiding*, M. Kirchner and D. Ghosal, Eds., Berlin, Heidelberg, 2013, pp. 160–175, Springer Berlin Heidelberg.

[42] J. Classen, M. Schulz, and M. Hollick, "Practical covert channels for WiFi systems," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Sep. 2015, pp. 209–217.

[43] K. S. Subraman, A. Antonopoulos, A. A. Abotabl, A. Nosratinia, and Y. Makris, "Demonstrating and mitigating the risk of an FEC-based hardware trojan in wireless networks," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 10, pp. 2720–2734, Feb. 2019.

[44] Y. Jin and Y. Makris, "Hardware trojans in wireless cryptographic ICs," *IEEE Design Test Comput.*, vol. 27, no. 1, pp. 26–35, Jan./Feb. 2010.

[45] Y. Liu, Y. Jin, A. Nosratinia, and Y. Makris, "Silicon demonstration of hardware trojan design and detection in wireless cryptographic ICs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 4, pp. 1506–1519, Apr. 2017.

[46] S. Chang, G. Bhat, U. Ogras, B. Bakkaloglu, and S. Ozev, "Detection mechanisms for unauthorized wireless transmissions," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 23, no. 6, pp. 70:1–70:21, Nov. 2018.

[47] K. S. Subramani, N. Helal, A. Antonopoulos, A. Nosratinia, and Y. Makris, "Amplitude-modulating analog/RF hardware trojans in wireless networks: Risks and remedies," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3497–3510, Apr. 2020.

[48] X. Zhang and M. Tehranipoor, "Case study: Detecting hardware trojans in third-party digital IP cores," in *IEEE Int. Symp. Hardw.-Oriented Secur. Trust (HOST)*, Jan. 2011, pp. 67–70.

[49] R. S. Chakraborty, F. Wolff, S. Paul, C. Papachristou, and S. Bhunia, *MERO: A Statistical Approach for Hardware Trojan Detection*, Berlin, Germany: Springer, 2009.

[50] V. R. Surabhi *et al.*, "Hardware trojan detection using controlled circuit aging," *IEEE Access*, vol. 8, pp. 77415–77434, Apr. 2020.

[51] M. Hicks, M. Finnicum, S. T. King, M. M. K. Martin, and J. M. Smith, "Overcoming an untrusted computing base: Detecting and removing malicious hardware automatically," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2010, pp. 159–172.

[52] A. C. Myers and B. Liskov, "A decentralized model for information flow control," in *Proc. 16th ACM Symp. Operating Syst. Princ. (SOSP)*, Oct. 1997, p. 129–142.

[53] X. Li *et al.*, "Sapper: A language for hardware-level security policy enforcement," *Proc. 19th Int. Conf. Archit. Support Program. Lang. Operating Syst. (ASPLOS)*, vol. 42, no. 1, pp. 97–112, Feb. 2014.

[54] Y. Jin, X. Guo, R. G. Dutta, M.-M. Bidmeshki, and Y. Makris, "Data secrecy protection through information flow tracking in proof-carrying hardware IP—part I: Framework fundamentals," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 10, pp. 2416–2429, Oct. 2017.

[55] M-M. Bidmeshki, X. Guo, R. G. Dutta, Y. Jin, and Y. Makris, "Data secrecy protection through information flow tracking in proof-carrying hardware IP—part II: Framework automation," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 10, pp. 2430–2443, Oct. 2017.

[56] K. Xiao, D. Forte, and M. Tehranipoor, "A novel built-in self-authentication technique to prevent inserting hardware trojans," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 33, no. 12, pp. 1778–1791, Dec. 2014.

[57] K. Shamsi, M. Li, K. Plaks, S. Fazzari, D. Z. Pan, and Y. Jin, "IP protection and supply chain security through logic obfuscation: A systematic overview," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 24, no. 6, pp. 65:1–65:36, Sep. 2019.

[58] A. Chakraborty *et al.*, "Keynote: A disquisition on logic locking," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 10, pp. 1952–1972, Oct. 2020.

[59] J. Rajendran, M. Sam, O. Sinanoglu, and R. Karri, "Security analysis of integrated circuit camouflaging," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, Nov. 2013, pp. 709–720.

[60] A. Vijayakumar, V. C. Patil, D. E. Holcomb, C. Paar, and S. Kundu, "Physical design obfuscation of hardware: A comprehensive investigation of device and logic-level techniques," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 1, pp. 64 – 77, Jan. 2017.

[61] Y. Wang, P. Chen, J. Hu, G. Li, and J. Rajendran, "The cat and mouse in split manufacturing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 5, pp. 805–817, May 2018.

[62] T. D. Perez and S. Pagliarini, "A survey on split manufacturing: Attacks, defenses, and challenges," *IEEE Access*, vol. 8, pp. 184013–184035, Oct. 2020.

[63] T. Sugawara *et al.*, "Reversing stealthy dopant-level circuits," *J. Cryptograph. Eng.*, vol. 5, no. 2, pp. 85–94, Jun. 2015.

[64] F. Stellari, P. Song, A. J. Weger, J. Culp, A. Herbert, and D. Pfeiffer, "Verification of untrusted chips using trusted layout and emission measurements," in *Proc. IEEE Int. Symp. Hardw.-Oriented Secur. Trust*, May 2014, pp. 19–24.

[65] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, "Trojan detection using IC fingerprinting," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2007, pp. 296–310.

[66] Y. Jin and Y. Makris, "Hardware trojan detection using path delay fingerprint," in *IEEE Int. Workshop Hardw.-Oriented Secur. Trust*, Jun. 2008, pp. 51–57.

[67] G. Bloom, B. Narahari, and R. Simha, "OS support for detecting trojan circuit attacks," in *Proc. IEEE Int. Workshop Hardw. Oriented Secur. Trust (HOST)*, Jul. 2009, pp. 100–103.

[68] S. Narasimhan, W. Yueh, X. Wang, S. Mukhopadhyay, and S. Bhunia, "Improving IC security against trojan attacks through integration of security monitors," *IEEE Des. Test Comput.*, vol. 29, no. 5, pp. 37–46, Sep./Oct. 2012.

[69] Y. Cao, C.-H. Chang, and S. Chen, "Cluster-based distributed active current timer for hardware trojan detection," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2013, pp. 1010–1013.

[70] Y. Jin and D. Sullivan, "Real-time trust evaluation in integrated circuits," in *Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE)*, Mar. 2014.

[71] C. Bao, D. Forte, and A. Srivastava, "Temperature tracking: Toward robust run-time detection of hardware trojans," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 10, pp. 1577–1585, Apr. 2015.

[72] S. Lapeyre, N. Valette, M. Merandat, M.-L. Flottes, B. Rouzeyre, and A. Virazel, "A plug and play digital ABIST controller for analog sensors in secure devices," in *Proc. 26th IEEE Eur. Test Symp. (ETS)*, May 2021.

[73] Y. Liu, G. Volanis, K. Huang, and Y. Makris, "Concurrent hardware trojan detection in wireless cryptographic ICs," in *Proc. IEEE Int. Test Conf. (ITC)*, Oct. 2015.

[74] X. T. Ngo, J.-L. Danger, S. Guilley, Z. Najm, and O. Emery, "Hardware property checker for run-time hardware trojan detection," in *Proc. Eur. Conf. Circuit Theory Design (ECCTD)*, Aug 2015.

[75] J.-L. Danger, L. Fribourg, U. Kühne, and M. Naceur, "LAOCOÖN: A run-time monitoring and verification approach for hardware trojan detection," in *Proc. 22nd Euromicro Conf. Digital Syst. Design (DSD)*, 2019, pp. 269–276.

[76] Y. Hou, H. He, K. Shamsi, Y. Jin, D. Wu, and H. Wu, "On-chip analog trojan detection framework for microprocessor trustworthiness," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 38, no. 10, pp. 1820–1830, Oct. 2019.

[77] J. Dworak, A. Crouch, J. Potter, A. Zygmontowicz, and M. Thornton, "Don't forget to lock your SIB: hiding instruments using P1687," in *Proc. IEEE Int. Test Conf. (ITC)*, Sep. 2013.

[78] R. Baranowski, M. A. Kochte, and H. Wunderlich, "Fine-grained access management in reconfigurable scan networks," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 6, pp. 937–946, Jun. 2015.

[79] S. Kan, J. Dworak, and J. G. Dunham, "Echeloned IJTAG data protection," in *Proc. IEEE Asian Hardw.-Oriented Secur. Trust*, Dec. 2016.

[80] J. Wang, C. Shi, A. Sanabria-Borbon, E. Sánchez-Sinencio, and J. Hu, "Thwarting analog IC piracy via combinational locking," in *Proc. IEEE Int. Test Conf. (ITC)*, Oct. 2017.

[81] N. G. Jayasankaran, A. S. Borbon, E. Sanchez-Sinencio, J. Hu, and J. Rajendran, "Towards provably-secure analog and mixed-signal locking against overproduction," in *Proc. 18th Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2018.

[82] V. Rao and I. Savidis, "Performance and security analysis of parameter-obfuscated analog circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 29, no. 12, pp. 2013–2026, Dec. 2021.

[83] J. Leonhard *et al.*, "Digitally-assisted mixed-signal circuit security," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 2021, early access.

[84] M. Elshamy, A. Sayed, M.-M. Louërat, H. Aboushady, and H.-G. Stratigopoulos, "Locking by untuning: A lock-less approach for analog and mixed-signal IC security," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 29, no. 12, pp. 2130–2142, Dec. 2021.

[85] A. Ash-Saki and S. Ghosh, "How multi-threshold designs can protect analog IPs," in *Proc. IEEE Int. Conf. Comput. Design (ICCD)*, Oct. 2018, pp. 464–471.

[86] J. Leonhard, A. Sayed, M.-M. Louërat, H. Aboushady, and H.-G. Stratigopoulos, "Analog and mixed-signal IC security via sizing camouflaging," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 40, no. 5, pp. 822–835, Jul. 2021.

[87] M. M. Bidmeshki, A. Antonopoulos, and Y. Makris, "Proof-carrying hardware-based information flow tracking in analog/mixed-signal designs," *IEEE J. Emerging Selected Topics Circuits Syst.*, vol. 11, no. 2, pp. 415–427, Jun. 2021.

[88] A. Coyette, B. Esen, W. Dobbelaere, R. Vanhooren, and G. Gielen, "Automatic generation of test infrastructures for analog integrated circuits by controllability and observability co-optimization," *Integration*, vol. 55, pp. 393–400, Sep. 2016.

[89] S. Sunter, J.-F. Côté, and J. Rearick, "Streaming access to ADCs and DACs for mixed-signal ATPG," *IEEE Des. Test. Comput.*, vol. 33, no. 6, pp. 38–45, Dec. 2016.