Circuit-to-Circuit Attacks in SoCs via Trojan-Infected IEEE 1687 Test Infrastructure

Michele Portolan, Antonios Pavlidis, Giorgio Di Natale, Eric Faehn, Haralampos-G. Stratigopoulos

Circuit-to-Circuit Attacks in SoCs via Trojan-Infected IEEE 1687 Test Infrastructure

Michele Portolan*, Antonios Pavlidis†, Giorgio Di Natale*, Eric Faehn‡ and Haralampos-G. Stratigopoulos†
*Université Grenoble Alpes, CNRS, Grenoble INP, TIMA F-38000, Grenoble, France
†Sorbonne Université, CNRS, LIP6, Paris, France
‡ST Microelectronics, Crolles, France

***Abstract*—We demonstrate a Hardware Trojan (HT)-based circuit-to-circuit attack mechanism in the context of Systems-on-Chip (SoCs). The HT trigger is hidden inside the attacking circuit and the HT payload travels from the attacking circuit to the victim circuit via the test infrastructure. The common test infrastructure is configured accordingly by the HT so as to propagate the HT payload. We demonstrate the capability of this HT to perform a denial-of-service attack on an industrial Analog-to-Digital Converter (ADC) connected to a IEEE 1687 test infrastructure.**

I. Introduction

A Hardware Trojan (HT) is a malicious modification of an Integrated Circuit (IC). It is composed of a trigger and a payload mechanism. The trigger defines the activation time (i.e., always-on, when a rare condition is met, time-based, external) and the payload is the effect of the activated HT on the victim IC (i.e., information leakage, degraded performance, denial of service). A HT can be inserted in any stage of the design process and at any level of abstraction and can be located anywhere on the die [1].

From the attacker's perspective, the goal is to make the HT stealthy and low-footprint so as to evade detection. HT designs are becoming increasingly sophisticated [2]–[4] making the development of countermeasures very challenging. Countermeasures include pre-silicon prevention of HT insertion (i.e., based on functional filler cells [5], logic obfuscation [6], camouflaging [7], or split manufacturing [8]), detection of the presence of HTs prior to IC usage (i.e., based on logic testing tools [9], Information Flow Tracking (IFT) [10], and side-channel analysis [11], [12]), and detection of HT activation during run-time (i.e., based on on-chip monitors [13]).

In this paper, we demonstrate a HT design implementing a circuit-to-circuit attack inside a System-on-Chip (SoC) by exploiting the Design-for-Test (DfT) infrastructure. The HT is hidden inside an "attacking" Intellectual Property (IP) core of the SoC and once activated it generates the payload in the form of a malicious bit pattern. The payload enters the scan chain of the test access mechanism, which traverses the SoC and controls the test instruments embedded inside the IPs. The HT manipulates the scan chain to propagate the payload at the interface of the target victim IP. The payload updates the status of the test instruments inside the victim IP, setting it to a partial and undocumented test mode, thus corrupting its functionality during normal operation mode. The circuit-to-circuit HT attack belongs to the broader family of scan attacks [14]. The principle of operation was originally proposed in [4]; however, how the HT manipulates the DfT infrastructure was only sketched without being implemented. In this paper, we show such an implementation for a DfT infrastructure based on the IEEE 1687 standard [15]. Our victim IP case study is an industrial Successive Approximation Register (SAR) Analog-to-Digital Converter (ADC).

The rest of the paper is organized as follows. In Section II, we describe the circuit-level implementation of the circuit-to-circuit HT attack. In Section III, we demonstrate the attack on the SAR ADC. In Section IV, we discuss the threat model and possible countermeasures. Finally, Section V provides conclusions and perspectives.

II. Hardware Trojan Design

A. Overview of IEEE 1687 standard

The main motivation behind the development of the IEEE 1687 standard [15] was the growing need of efficient access to embedded test instruments in ICs. In fact, while scan chains offer an easy access to chip internals, their length makes the usage for embedded test instruments cumbersome and sub-optimal. This was solved by the IEEE 1687 standard by introducing the Segment Insertion Bit (SIB), an element that allows to add or remove segments of the scan chain depending on its status ('0' or '1'). It is therefore possible to create hierarchical levels as depicted in Fig. 1, where the box on the left is the IEEE 1149.1 Test Access Port (TAP) controller [16]. A new Instrument Connectivity Language (ICL) is introduced to describe these new scan chain topologies, along with a new Procedural Description Language (PDL) to express functional operations of test instruments.

One of the main strengths of the IEEE 1687 standard is that it supports a multi-actor flow. Each IP provider can deliver the design together with its own IEEE 1687 standard elements described in ICL. Then, the composition of the DfT infrastructure is decided and test programs are re-targeted to the top-level design. The IEEE 1687 standard is now in full swing, and major Electronic Design Automation (EDA) providers support it in their tool suites.

B. IEEE 1687-capable HT Design

The circuit-to-circuit HT attack will be described with the help of Fig. 2. As any trigger mechanism can be used inside the attacking IP, we focus only on the payload mechanism, i.e., how the HT manipulates the test infrastructure. A small Finite

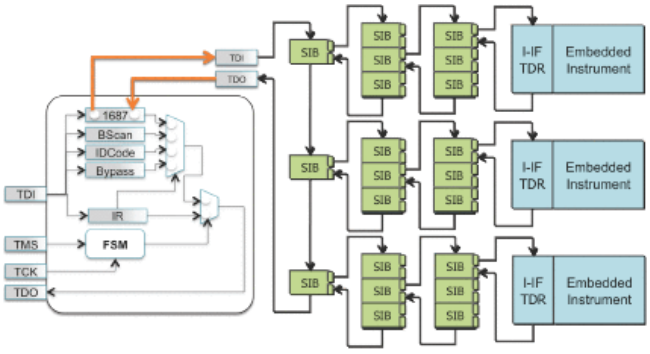
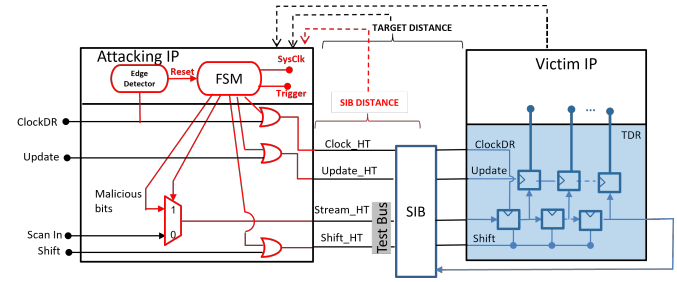
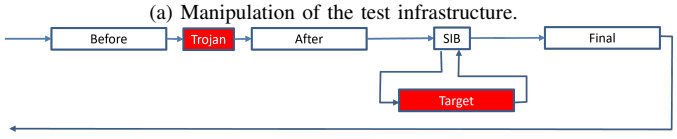


Fig. 1: Example of SIB-based hierarchy (from [17]).



(a) Manipulation of the test infrastructure.



(b) Hierarchical view.

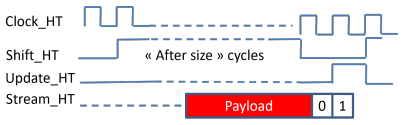
Fig. 2: HT design for a SIB-based topology.

State Machine (FSM) inside the attacking IP orchestrates the attack. Upon triggering, it issues the payload and overrides the IJTAG control and data signals. The payload enters the *scan in* path and a *shift* operation is held until the payload reaches and aligns with the Test Data Register (TDR) of the victim IP. At this moment, an *update* operation loads the payload onto the test interface of the victim IP completing the attack.

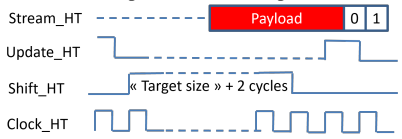
With respect to [4], three main modifications are made:

- In most implementations, the scan chain clock signal *ClockDR* is derived from the clock applied to the TAP interface. *ClockDR* should not be used for the HT attack to run outside normal scan operation. We therefore exploit the system clock *SysClk*, and generate an internal clock through bit-banging. This is a widespread technique that leverages the low frequencies of the scan interface (usually between 10 and 100MHz) with respect to the system's frequency (usually several hundred MHz).
- We monitor activity on the *ClockDR* using an edge detector to immediately react to any operation on the TAP interface. In this way, if the circuit is set to test mode, then the HT payload is by construction suppressed making the HT stealthy during test mode.
- An attack protocol is defined to deal with the variable length scan chain enabled by SIBs.

The last point is described next in more detail. Let us consider Fig. 2(a), where access to the TDR of the victim IP



(a) Step 1: force SIB Open.



(b) Step 2: Deliver Pattern.

Fig. 3: Attack protocol.

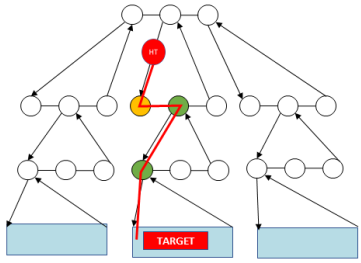


Fig. 4: Example of HT insertion in the system of Fig. 1.

is controlled by a SIB. Fig. 2(b) provides an hierarchical view. The TDR of the victim IP belongs to a different hierarchical level than the HT and is accessible only when its SIB is open, i.e., its control bit is set to '1'. The SIB distance is the size of the 'After' scan chain segment, and the distance to the victim IP is one additional bit to account for the SIB control register. Therefore, the HT needs first to open the SIB before driving the payload into the TDR of the victim IP. In the general case, the network topology may present a deeper hierarchy as shown in Fig. 1. Therefore, SIBs need to be sequentially opened or closed accordingly to reach the victim IP. This is done by pre-pending a '01' sequence to the payload, as shown in the attack protocol of Fig. 3. This two-bit word is used as a "chisel" to manipulate the encountered SIBs, i.e., a number of shift cycles equal to the SIB distance is performed so as to align a '0' or '1' to force closing or opening a SIB with an update cycle, respectively. An example is shown in Fig. 4, where the test setup Fig. 1 is redrawn as a tree using SIBs as nodes and TDRs as leaves. Given the position of the HT and victim IP in the hierarchy, a shortest path algorithm can be employed to reach the victim IP. In Fig. 4, the *Shift-Update* (SU) cycle needs to be reproduced three times to respectively close-open-open the three SIBs encountered. Thereafter, we resume shifting for a number of cycles needed to align the payload with the TDR of the victim IP, and an update is issued to apply the payload and complete the attack. All the information needed to build this hierarchical representation and define the attack protocol is in the ICL files at the disposal of the attacker.

Note that the FSM in the IEEE 1149.1 TAP controller [16] dictates a *Capture-Shift-Update* (CSU) cycle: all data are updated simultaneously after shifting is over, so all SIBs are updated together. Data alignment is paramount and depends on both the network topology and the current state. It is one of the

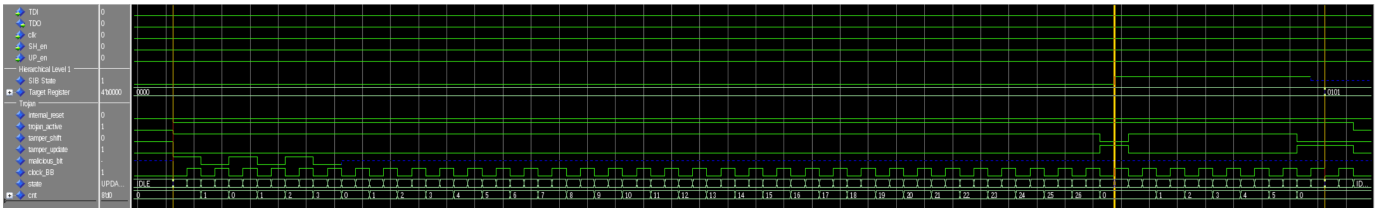
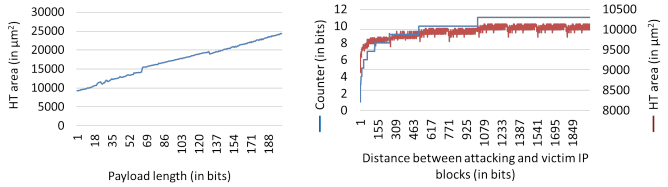


Fig. 5: RTL simulation of the HT attack of Fig. 2.



(a) HT area vs. payload length. (b) HT area vs. distance between the attacking and victim IPs for a 4-bit payload.

Fig. 6: HT size.

most difficult tasks assigned to the retargeter tool. However, the proposed HT does not suffer from this complexity. It generates its own scan control signals through bit-banging, thus it does not need to comply to the CSU cycle. Shift and update operations can be mixed freely and used repeatedly. The HT purposefully generates update cycles to force the value of the SIBs encountered while shifting the payload.

Fig. 5 shows an RTL simulation of the attack using the test setup of Fig. 2. The fist vertical yellow cursor on the left shows the activation of the HT, which starts shifting the chisel and payload. When the chisel is aligned with the SIB (second cursor), an update cycle is generated to open it. Shift is then resumed until the desired payload reaches the victim IP and can update the status of the test instruments (last cursor).

C. HT Size

A critical parameter for a HT is size: the larger its footprint is on the die, the easier detection will be. As the circuit-to-circuit HT attack can make use of any state-of-the-art trigger mechanism, herein we focus on the rest of the HT design, shown in red color in Fig. 2. To assess its size, we synthesised it in the ams 0.35μm CMOS technology while varying its main parameters, namely the payload length and the distance between the attacking and victim IPs.

Fig. 6(a) shows the total area of the HT as a function of the payload length. The impact of the payload length is extremely limited, especially considering that in most applications it will be rather short. The size of the HT itself is also extremely small, i.e., around $10 - 25K\mu\text{m}^2$. To make a comparison, in the same technology, a 1-bit boundary scan cell occupies an area close to $1.5K\mu\text{m}^2$, while an 8-bit multiplier goes up to $20K\mu\text{m}^2$. In the scale of a full SoC, the HT will be effectively drowned within the functional logic and, thereby, it will be difficult to be distinguished from legitimate hardware.

The distance between the attacking and victim IPs has a larger impact, as depicted in Fig. 6(b). The larger the distance is, the bigger the counter inside the FSM needs to be so as to guarantee a correct alignment of the payload with the TDR of the victim IP, with the increase itself being logarithmic. Nevertheless, the HT area remains extremely small even when the distance to the target is very high, thus leaving great freedom to the attacker in choosing the HT placement.

III. HT Attack Demonstration on a SAR ADC

A. SAR ADC Architecture and BIST Infrastructure

As a victim IP we consider a 65nm CMOS SAR ADC by ST Microelectronics. The SAR ADC top-level architecture along with its BIST interface to the common SoC test infrastructure are shown in Fig. 7. The circuit has two separate Built-in Self-Test (BIST) mechanisms embedded, namely Symmetry-based BIST (*SymBIST*) [18] and topology modifications [19], both having a fully-digital interface being accessed and controlled by the SoC test infrastructure.

SymBIST is an one-fits-all BIST paradigm for Analog and Mixed-Signal (AMS) ICs that can be reused for defect-oriented post-manufacturing testing [18], on-line concurrent error detection [18], and fault diagnosis [20]. It is based on identifying or constructing invariances into the design, where an invariance is a signal that by default in fault-free operation stays within a tolerance window for any input. In abnormal operation, however, one or more invariances are violated, i.e., the signal slides outside the tolerance window. Each invariance is generated and checked for compliance by a dedicated on-die checker. The checker outputs '0' if the invariance is satisfied and '1' if it is violated. The outputs of the checkers are combined driving an AND gate so as to generate a single 1-bit pass/fail decision. For the SAR ADC, all invariances are located inside the main SAR cell that performs the conversion algorithm. The reason is that signals generated by all other sub-blocks are all processed by the SAR cell and, thereby, they will be affecting one or more invariances inside the SAR cell. The *SymBIST* infrastructure comprises in addition two internal test stimulus generators which are employed in the defect-oriented post-manufacturing testing and diagnosis phases.

Topology modifications are enabled by connecting single transistors to nodes of the design. When a transistor is activated, the node is pulled to ground (for a NMOS transistor) or to V_{DD} (for a PMOS transistor). By activating any combination of transistors, the topology of the design is modified. The underlying idea is that topology modifications can amplify the effect of a defect making it observable. For the SAR ADC, topology modifications are introduced only inside the bandgap for improving defect coverage and diagnosis

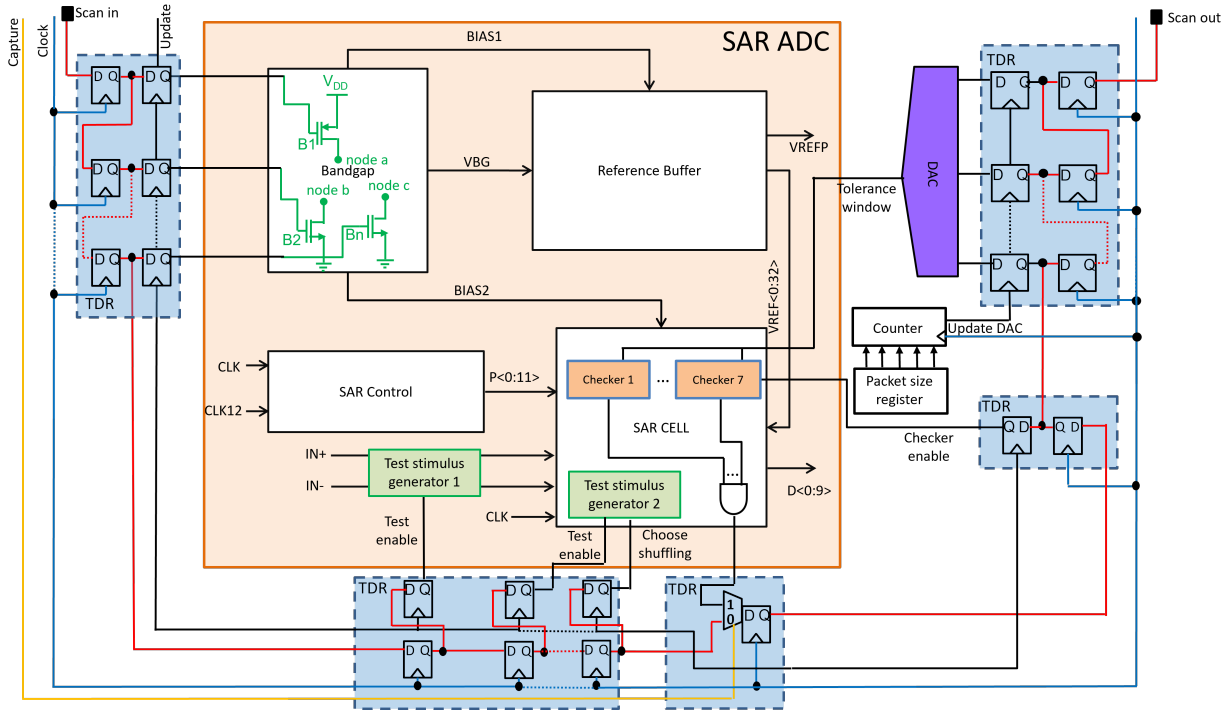


Fig. 7: SAR ADC top-level architecture showing its BIST interface to the SoC test infrastructure.

resolution. *SymBIST* is used to readout a test signature for each topology modification.

As shown in Fig. 7, the SAR ADC BIST control signal is composed of: (a) a digital word that controls the gates of the topology modification transistors inside the bandgap; (b) a digital word that enables and controls the test stimulus generators; (c) a digital word that enables the checkers; and (d) a digital word that when converted with a Digital-to-Analog Converter (DAC) generates the thresholds used in a checker to set its tolerance window.

The interested reader is referred to [18], [20] for a more detailed description of the SAR ADC IP and its BIST infrastructure.

B. HT Payload

The HT payload consists in switching on the BIST or part of it during normal operation. We demonstrate herein one scenario where the HT payload switches on one topology modification transistor inside the bandgap. Fig. 8 shows various signals of interest. In this example, the input is a decaying sinusoidal and the HT is triggered at around 10.8μs. When the HT is triggered and the payload reaches and updates the BIST control signal, this re-configures the bandgap generating incorrect bias voltages, as shown in the third subplot of Fig. 8. In turn, other sub-blocks that are supplied by the bandgap get infected as well. The last subplot in Fig. 8 shows the effect on the SAR ADC output. The output codes immediately deviate from their expected HT-free values. Thus, the SAR ADC produces wrong input digitization upon HT activation. We observe that when the HT trigger is off, the circuit returns to normal operation with some noticeable delay.

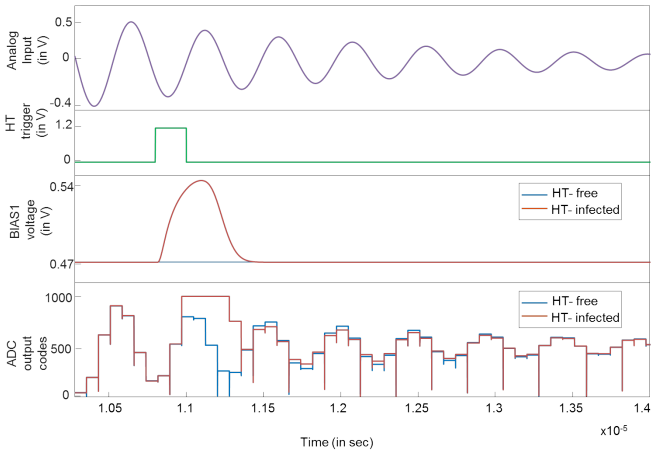


Fig. 8: HT payload turns on a topology modification transistor inside the bandgap.

IV. Threat Model and Security Analysis

The circuit-to-circuit HT attack is general: any two IPs in a SoC can play the role of the attacking and victim IPs. The attack can be implemented at the final design stage by the system integrator. The system integrator will need to insert the HT in the DfT before the victim IP and connect it to the trigger mechanism of his choice. The HT is configured based on the ICL files. The HT could also be implanted by a malicious foundry by modifying the layout in the GDSII file.

Generic HT prevention and detection countermeasures, such as those mentioned in Section I, could in principle be used to defend against the circuit-to-circuit HT attack. The detailed evaluation of their effectiveness and cost is out of the scope of this paper. It is worth mentioning that test-based

countermeasures will fail since, as discussed in Section II-B, when the circuit is in test mode the HT payload mechanism is invalidated. Thus, the HT effect will not be measurable as a way of detection. Also, it is worth mentioning that *SymBIST* was demonstrated recently as an effective run-time HT detection technique for AMS ICs [13].

Another family of countermeasures is related to gaining trust in DfT infrastructures [14]. Existing approaches include access authentication [17], [21], assuring data confidentiality and integrity [22], and on-line detection of test pattern compliance [23]. Among all, the most famous approach is the Locking SIB (LSIB) [17]. The idea is to regroup sensitive areas of the scan chain in a given segment behind a special SIB whose opening depends for instance on providing a secret key to a specific location in the DfT topology. The secret key is not directly communicated to the system integrator. However, as it is static, an analysis of the ICL and/or the circuit could allow a malicious system integrator to extract it [24]. Thereafter, it can be simply treated as a pattern to be delivered by the HT, similar to the pattern that opens a SIB as explained in Section II-B, before the actual payload is delivered. Thus, the circuit-to-circuit HT attack can bypass the LSIB defense. One potential countermeasure would be to use dynamic keys whose validity is limited to a given session. Proposals exist to provide such security, for instance using a challenge-response scheme to control a LSIB [25]. However, these approaches incur an important overhead cost, so their actual usage is still limited. Moreover, they are not part of the IEEE 1687 design flow, thus they need custom flows in addition to EDA tools. Proposals to include such constructs into the standard flow do exist [26], but so far no commercial tool implements them.

V. Conclusions

We presented an implementation of the circuit-to-circuit HT attack for a DfT infrastructure based on the IEEE 1687 standard. We demonstrated the HT payload for a SAR ADC. Yet, the victim IP can be any IP in a SoC. The HT has a tiny footprint and its payload mechanism is disabled during testing, thus making it stealthy against test-based defenses. Appropriate defenses need to be developed focusing on increasing the trust of the DfT infrastructure itself. Other future work directions will include demonstrating circuit-to-circuit HT payloads for other circuit classes and continuing developing the HT design for more complex DfT topologies.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] S. K. Haider, C. Jin, M. Ahmad, D. M. Shila, O. Khan, and M. van Dijk, "Advancing the state-of-the-art in hardware trojans detection," *IEEE Trans. Dependable Secure Comput.*, vol. 16, no. 1, pp. 18–32, Jan./Feb. 2019.

[10] 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.

[11] J. He, Y. Zhao, X. Guo, and Y. Jin, "Hardware trojan detection through chip-free electromagnetic side-channel statistical analysis," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 10, pp. 2939–2948, Oct. 2017.

[12] A. Stern, D. Mehta, S. Tajik, U. Guin, F. Farahmandi, and M. Tehranipoor, "SPARTA-COTS: A laser probing approach for sequential trojan detection in COTS integrated circuits," in *IEEE Phys. Assur. Insp. Electron. (PAINE)*, Dec. 2020.

[13] A. Pavlidis, E. Faehn, M.-M. M. Louërat, and H.-G. Stratigopoulos, "Run-time hardware trojan detection in analog and mixed-signal ICs," in *Proc. IEEE VLSI Test Symp. (VTS)*, Apr. 2022.

[14] 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.

[15] "IEEE standard for access and control of instrumentation embedded within a semiconductor device," *IEEE Std 1687-2014*, pp. 1–283, 2014.

[16] "IEEE standard for test access port and boundary-scan architecture," *IEEE Std 1149.1-2013 (Revision of IEEE Std 1149.1-2001)*, pp. 1–444, 2013.

[17] 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.

[18] 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.

[19] A. Coyette, B. Esen, R. Vanhooren, W. Dobbelaere, and G. Gielen, "Automated testing of mixed-signal integrated circuits by topology modification," in *Proc. IEEE VLSI Test Symp. (VTS)*, Apr. 2015.

[20] 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.

[21] 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.

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

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

[24] N. Lylina, C.-H. Wang, and H.-J. Wunderlich, "SCAR: Security compliance analysis and resynthesis of reconfigurable scan networks," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, 2022, early access.

[25] M. Portolan, V. Reynaud, P. Maistri, and R. Leveugle, "Dynamic authentication-based secure access to test infrastructure," in *Proc. IEEE Eur. Test Symp. (ETS)*, 2020.

[26] M. Portolan, V. Reynaud, P. Maistri, R. Leveugle, and G. Di Natale, "Security EDA extension through P1687.1 and 1687 callbacks," in *Proc. IEEE Int. Test Conf. (ITC)*, Oct. 2021, pp. 344–353.