Locking by Untuning: A Lock-Less Approach for Analog and Mixed-Signal IC Security

Mohamed Elshamy, Alhassan Sayed, Marie-Minerve Louërat, Hassan Aboushady, Haralampos-G. Stratigopoulos

Locking by Untuning: A Lock-Less Approach for Analog and Mixed-Signal IC Security

Mohamed Elshamy, Alhassan Sayed, Marie-Minerve Louërat, Hassan Aboushady, *Senior Member, IEEE*, and Haralampos-G. Stratigopoulos, *Member, IEEE*

***Abstract*—We propose an anti-piracy security approach for programmable analog and mixed-signal (AMS) Integrated Circuits (ICs). The security approach relies on functionality locking by leveraging the inherent programmability and utilizing the configuration settings as secret keys or, equivalently, the programming bits as key-bits. When invalid keys are applied, the circuit is untuned and, as a result, its functionality breaks, i.e., at least one of the performances violates its specification. As long as the calibration algorithm that produces the configuration settings can be kept secret, the proposed approach can serve as a countermeasure against all types of counterfeiting, i.e., cloning, overbuilding, remarking, and recycling. An important advantage of the proposed approach is that it is lock-less. It leaves the design intact, there is no change to the design flow, and there is no performance penalty and no area or power overheads due to the lock operation. We demonstrate it on a $\Sigma\Delta$ Analog-to-Digital Converter (ADC) with 194-bit programmability and complex calibration algorithm used in the context of highly-digitized, multi-standard RF receivers.**

***Index Terms*—Hardware security and trust, mixed-signal integrated circuits, IP/IC piracy, locking, tuning, calibration.**

I. Introduction

Hardware security and trust of Integrated Circuits (ICs) and Intellectual Property (IP) blocks in Systems-on-Chip (SoCs) is a topic that has attracted a lot of interest in recent years. There are various threats, including IP/IC piracy, hardware Trojans, side-channel attacks, and fault injection attacks [1]. In this work, we focus on IP/IC piracy and we propose an anti-piracy countermeasure for analog and mixed-signal (AMS) ICs that embed digitally-controlled programmability.

IC/IP piracy includes reverse engineering [2], [3] and different counterfeiting types [4], namely cloning, overbuilding, remarking, and recycling. The first three counterfeiting types are a corollary of the horizontal semiconductor manufacturing business model where many semiconductor companies outsource IP design, IC fabrication, and IC testing to multiple potentially untrusted entities with the goal to reduce costs.

Manuscript received May 8, 2021; revised July 30, 2021, and August 31, 2021; accepted September 17, 2021. This work was supported by the ANR STEALTH project under Grant ANR-17-CE24-0022-01 and by the RAPID FLEXyRADIO project. This article was recommended by Associate Editor T.-Y. Ho. (Corresponding author: Haralampos-G. Stratigopoulos.)

Mohamed Elshamy, Marie-Minerve Louërat, Hassan Aboushady, and Haralampos-G. Stratigopoulos are with the Sorbonne Université, CNRS, LIP6, 75005 Paris, France (e-mail: mohamed.elshamy@lip6.fr; marie-minerve.louerat@lip6.fr hassan.aboushady@lip6.fr; haralampos.stratigopoulos@lip6.fr).

Alhassan Sayed is with the Sorbonne Université, CNRS, LIP6, 75005 Paris, France, and also with the Electronics and Communications Department, Minia University, Minia 61519, Egypt (e-mail: alhassan.sayed@lip6.fr).

Digital Object Identifier 10.1109/TVLSI.2021.XXXXXXX

Reverse engineering refers to the procedure of deriving IP/IC proprietary information, i.e., architecture, netlist, layout, etc., from a fabricated chip. Nowadays, there exist equipment and software tools to successfully reverse-engineer any unprotected chip [2], [3]. The procedure involves the following steps: (a) de-packaging of the chip; (b) de-layering the individual layers of the die using corrosive chemicals; (c) imaging the top-view of each layer using, for example, Scanning Electron Microscopy (SEM); (d) aligning and stitching the images of the different layers; and (e) extracting the netlist from the annotated images using dedicated software tools. An adversary may perform reverse engineering to reduce its technological disadvantage against the "author" of the IP/IC, possibly gathering the necessary information for producing a similar or identical, i.e., cloned, IP/IC.

A cloned counterfeit is an IP/IC that is being illegally cloned and sold as original. Cloning can be performed by a third-party malicious SoC integrator or foundry that receives the blueprint of the IP/IC or by an adversary via reverse-engineering of a legally purchased chip. Overbuilding refers to extra ICs fabricated by a malicious foundry beyond the number agreed in the contract with the IC design house, which thereafter are illegitimately sold. Remarked ICs are failing ICs that are remarked by a malicious test facility as passing ICs and are sold with false and forged documentation. Recycling refers to de-soldering a dispensed used or defective board, extracting a chip, remarking it, and selling it in the market as new.

Nowadays, IP/IC piracy is a major preoccupation for governments, industry, and consumers. For governments it poses a national security threat, i.e., when counterfeits are used in critical infrastructure and defense. For industry it results in financial losses and damage to the brand value. For consumers it poses safety risks since counterfeited parts have lower reliability and also it results in replacement costs due to higher failure probability of counterfeited parts.

Countermeasures against IP/IC piracy have been extensively studied for digital ICs for over a decade now. Main countermeasures include split manufacturing [5], [6], physical design obfuscation [7], [8], and logic locking [9], [10].

Split manufacturing protects only against an untrusted foundry by manufacturing only a part of the target design at the untrusted high-end foundry and the remaining part at a trusted low-end foundry.

Physical design obfuscation protects only against reverse-engineering by making "stealthy" alterations in the design using mechanisms at the device and interconnect level, resulting in an extracted netlist that is "deceiving" for the attacker.

Locking, on the other hand, is an end-to-end protection mechanism. It consists of inserting a lock into the design such that unless the valid key is used the functionality breaks. The key is kept as the IP/IC design house secret. Locking thwarts cloning by the SoC integrator or foundry and overbuilding by the foundry as the blueprint of the IP/IC is useless without knowing the key. It also thwarts cloning via reverse-engineering as the key is stored in a Tamper-Proof Memory (TPM) and any attempt to read it will result in irreversible key loss. It thwarts remarking for digital ICs since structural testing can be equivalently performed on a locked chip using any invalid key and, thereby, chips can be unlocked after testing [11]. Protection against remarking can be achieved for any IC type by remotely activating the chips during testing using asymmetric cryptography [12]. Finally, it thwarts recycling as long as the key is reloaded every time the IC is powered-up. This requires a different key management scheme that makes use of a public user-key and a chip identification-key generated, for example, by an on-chip Physical Unclonable Function (PUF) [13]. The user-key and chip identification-key are XORed to produce the secret key.

Regarding AMS ICs, in general, hardware security and trust aspects are largely unexplored as of today [14]–[17]. The first anti-piracy methodologies have appeared recently. The vast majority of works focus on locking and can be broadly categorized into biasing locking [18]–[22], locking the on-chip calibration mechanism [23], [24], and system-level locking by leveraging logic locking of digital sections [25]–[27]. The principle of split manufacturing is demonstrated in [28]. It is argued that an untrusted foundry not knowing coil and capacitor sizings, grants a higher level of security of split manufacturing for RF than for digital designs. Existing physical design obfuscation techniques for AMS ICs are based on obfuscating the threshold voltage of transistors [29] and camouflaging the layout geometry of analog components [30].

In this work, we propose a locking methodology that applies to AMS ICs that embed multiple-bit digitally-controlled programmability. We argue that in this case a lock-less solution can be envisioned. In particular, we argue that the tuning knobs within the programmable analog IC can naturally serve as a locking mechanism. Specifically, the programming bits or digital codes controlling the tuning knobs serve as key-bits and each configuration setting, i.e., programming bits that configure the IC in a specific operation mode demanded by the application, is treated as a secret key. Naturally, when invalid programming bits are provided the functionality of the circuit breaks. A requirement for this locking scheme is that the calibration algorithm that produces the configuration settings is also kept secret. We discuss the practical implementation of this locking approach, its benefits, and its resilience against foreseen attacks. We demonstrate it with hardware measurements on a 194-bit programmable $\Sigma\Delta$ Analog-to-Digital Converter (ADC) used in highly-digitized, multi-standard RF receiver applications. Locking programmable AMS ICs via protecting their tuning settings and the underlying calibration algorithm was originally proposed in [31].

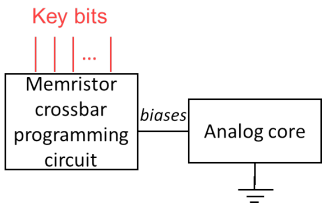
The existing AMS IC locking methodologies [18]–[26], which will be described in more detail in Section II, are generic and, thus, can also be used in the context of programmable AMS ICs. Their common characteristic is that they all insert a lock into the design. The lock is thoughtfully inserted into peripheral circuitry, i.e., biasing circuitry or on-chip calibration mechanism, or into digital sections with the aim to be as non-intrusive as possible to the sensitive analog core. These lock insertion approaches also offer the possibility to use a large key size, which is a prerequisite for thwarting counter-attacks aiming at recovering the key. However, as it will be discussed in Section II, these lock insertion approaches may require changes in the AMS design flow, may degrade performance requiring careful co-design with the target AMS IC, may result in some justifiable yet non-negligible area and power overheads, and some are also vulnerable to counter-attacks. We postulate that in the case of programmable ICs the proposed lock-less solution when applicable offers several comparative advantages: no circuit re-design or change in the design flow, no performance penalty, and no area or power overheads. Of course, this comparison does not hold true in general since in the absence of programmability the proposed solution is not applicable.

The rest of the paper is structured as follows. In Section II, we discuss previous work on AMS IC locking. In Section III, we provide a general overview of programmability embedded into AMS ICs. In Section IV, we present the proposed locking methodology and we discuss its benefits and its resilience against foreseen counter-attacks. In Section V, we describe our case study. In Section VI, we present the experimental results. Section VII concludes the paper.

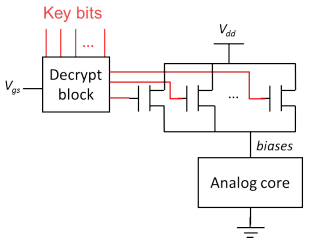
II. Prior art in locking AMS ICs

Existing methodologies for locking AMS ICs are illustrated in Fig. 1.

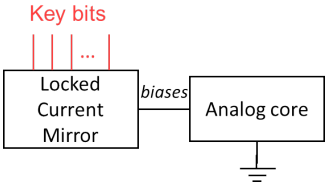
Biasing locking inserts the lock into the biasing circuits that provide the steady currents or voltages, i.e. biases, that set the quiescent operating conditions of the analog IC. The key controls the generation of biases. An invalid key will generate offset biases, thus affecting the intent performance trade-off. In [18], a key-enabled biasing circuit is proposed based on two memristor crossbars, as illustrated in Fig. 1a. Only the valid key programs correctly the memristor crossbars to generate the correct biases, while by applying an invalid key the memristor breakdown voltage can be reached, thus limiting the number of trials an attacker can attempt. In [19], a transistor that sets the current or voltage bias in a node is obfuscated by replacing it with parallel-connected transistors whose gates are controlled by key-bits, as illustrated in Fig. 1b. The key-bits activate transistors whose aggregate width equals the nominal width of the original transistor. In [20], a redesign of a current mirror is proposed so as to insert key-bits, as illustrated in Fig. 1c. Extra mirroring branches are inserted, where each branch is comprised of the mirroring transistor and possibly several switches that are controlled by the key-bits. The resultant current bias will depend on which branches are switched-on, as well as on the geometry of the mirroring transistor in these branches. In [21], a mesh-based obfuscation of biasing transistors is proposed, as illustrated in Fig. 1d. Each



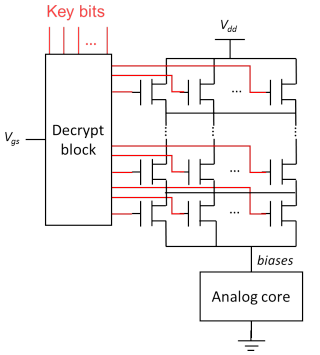
(a) Biasing locking based on memristor crossbar [18].



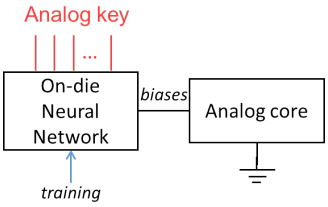
(b) Biasing locking based on transistor width obfuscation [19].



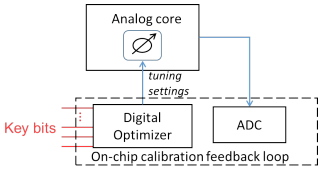
(c) Biasing locking based on current mirror locking [20].



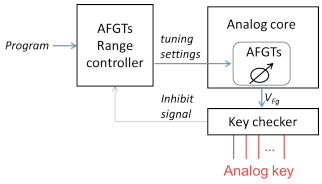
(d) Biasing locking using mesh-based transistor obfuscation [21].



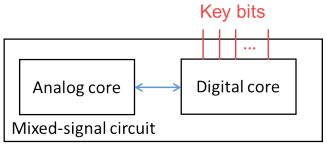
(e) Locking through neural network-based biasing [22].



(f) Logic locking of digital optimizer in on-chip calibration loop [23].



(g) Locking of the programming range of AFGTs [24].



(h) MixLock [25]–[27].

Fig. 1: Existing locking methodologies for AMS ICs.

transistor in the mesh is of different size and is controlled by a key-bit. The valid key sets the effective transistor length and width that generates the correct bias. In [22], it is proposed to add on-chip a neural network that is trained to map the secret analog key, which is in the form of analog DC voltages presented as inputs to the neural network, to the correct bias, as illustrated in Fig. 1e. For invalid keys the neural network is trained to give the same erroneous bias.

Biasing locking is a generic methodology applicable to any AMS IC and offers an elegant way for inserting a digital key-enabled lock into an AMS IC. Although the lock is not inserted into the analog core, biasing circuits are fundamental units for proper operation of AMS ICs and their design should be carefully done to meet requirements such as biasing accuracy, temperature stability, bandwidth, input/output compliance voltage, input/output resistance, etc. The above works have not considered the effect of locking on the performance of the biasing circuit.

On the other hand, counter-attacks were proposed recently that break biasing locking allowing the attacker to recover the secret key or to remove the lock by re-synthesizing the biasing circuit [32]–[35].

Another category of locking methodologies considers inserting the lock into the on-chip calibration mechanism. In this case, locking acts on the tuning knobs that compensate for process variations and non-idealities. In [23], a calibration loop is considered that uses an ADC to digitize the output of the circuit, followed by a digital optimizer that maps the output to appropriate tuning knobs that improve the performance trade-off of the circuit. It is proposed to insert the lock into the digital optimizer using logic locking, as illustrated in Fig. 1f. Unless the valid key is provided, the tuning operation is affected. Logic locking results in some justifiable yet non-negligible area and power overheads. In [24], a calibration scheme is considered enabled by Analog Floating-Gate Transistors (AFGTs). A locking principle is proposed where the lock controls the programmability range of the AFGTs, as illustrated in Fig. 1g. The full tuning range is inhibited unless the AFGTs are first programmed in a certain order and with certain voltages, which are termed waypoints and constitute the secret analog key. This condition is validated by a key checker block. The limitations of this approach are that AFGTs are not standard tuning knobs to enable programmability and that the lock mechanism can be straightforwardly removed by the attacker, thus this approach does not offer strong resilience against cloning.

The third locking methodology leverages logic locking of digital sections within an AMS IC to gain control over the signal-processing flow [25]–[27], as illustrated in Fig. 1h. Using an invalid key, the output of the locked digital sections will be corrupted, which will in turn corrupt the AMS IC performance trade-off in a complex and unpredictable way. Similar to [23], logic locking will result in some justifiable yet non-negligible area and power overheads.

Finally, a compound locking methodology can be considered. For example, in [36], it is proposed to lock the analog core with biasing locking and the digital core with logic locking creating shared key dependencies so as to stop an attacker from breaking the analog and digital cores' locking mechanism independently.

III. Programming of AMS ICs

AMS ICs are often demanded to be programmable (or configurable) with the aim to: (a) Compensate for process variations and inherent non-idealities so as to achieve the desired performance trade-off and boost yield; (b) Configure the circuit into different operation modes demanded by the application; (c) Adapt the performance to changes in the environment, e.g., towards moderating power consumption; (d) Enable fault tolerance in the presence of aging, latent defects, single event upsets, etc.

Programmability (or configuration) is enabled by judiciously inserting tuning knobs (or actuators) into the design that act on the circuit performances. Typically, tuning knobs are programmable bias sources that set the current of voltage bias in a node or are implemented by tunable single components, i.e., resistors, capacitors, and varactors. Ideally, tuning knobs should act orthogonally on the circuit performances so as to facilitate finding a good balance among multiple competing performance goals; however, this orthogonality property is difficult to achieve in practice and a tuning knob typically acts simultaneously on multiple performances invoking a trade-off, which makes the programming more tangled. Typically, tuning knobs are digitally-controlled, that is, a configuration setting is a digital word.

The programming is driven by a calibration algorithm that uses performance indicators, i.e., direct measurement of performances or information-rich measurements, to search in the space of tuning knob settings so as to achieve the target performance objectives. The calibration algorithm returns the configuration setting (or programming bits) that sets the optimal performance trade-off given the target objective. It is an optimization process that involves multiple testing/tuning iterations where in each iteration the next best tuning knob setting is selected based on the current trade-off of measured performances. The calibration algorithm can be implemented off-chip or on-chip.

Off-chip calibration requires that the chip is interfaced with the Automated Test Equipment (ATE). The ATE applies test stimuli, analyses the test response, and generates the tuning knob values. The calibration algorithm runs on the computer of the ATE. The tuning knobs are accessed and controlled from a primary pin via a test bus. The calibration can be driven directly by the measured performances or can be assisted by design-for-test (DfT) structures with the aim to reduce test cost, i.e., interface the chip to low-cost ATE and reduce the number of test configurations and/or test time. For example, in [37], a built-in envelope detector is used to extract the low-frequency envelope of the output of an RF transmitter from which multiple RF performances are predicted implicitly in a single test step using the alternate test principle. In [38], loop-back test is used to analytically compute the parameters of the RF transceiver using baseband test signals and, thereafter, these parameters are used for pre-distortion or post-distortion to digitally calibrate the RF transceiver. In [39]–[41], it is shown that by adding on-chip process variation-aware sensors that are not electrically connected to the circuit under calibration, the calibration can be performed in "one-shot" based on machine learning algorithms. An off-chip implementation of the calibration algorithm can address objectives (a)-(c). For objectives (b)-(c), operation modes and adaptation levels need to be pre-specified based on the anticipated range of applications and environmental conditions, resulting in multiple pre-specified configuration settings. In this case, the calibration algorithm returns a look-up table (LUT) with the multiple pre-specified configuration settings that is pre-loaded into the chip before deployment and stays fixed during its entire lifetime. Based on the application and the environmental conditions met in the field, the appropriate configuration setting is selected from the LUT.

An on-chip implementation [38], [42]–[46] requires an on-chip infrastructure that includes, for example, measurement acquisition sensors for obtaining performance indicators, ADCs for digitizing the measurements, digital post-processing circuitry that drives the tuning knob values optimization given the current measurements, and Digital-to-Analog Converters (DACs) if the actuating signals are analog. An on-chip calibration is automated and can be completed faster compared to off-chip calibration since the process takes place entirely on-chip and there is no need to offload test signals and perform off-chip analysis. An on-chip implementation can address all aforementioned objectives (a)-(d). For objectives (b)-(c), the LUT approach can be followed. For objective (c), in case where the changes in the environment cannot be anticipated, a fully embedded on-chip calibration scheme offers the possibility to run the calibration on-chip upon request. For objective (d), this is required since fault scenarios are manifold. Clearly, an on-chip implementation offers larger flexibility compared to an off-chip implementation. It can decide on the best configuration setting considering the current status of the chip. However, this comes at the expense of area overhead and design complexity, thus oftentimes it is not the preferred solution by designers.

In summary, both off-chip and on-chip calibration implementations can meet objectives (a)-(c). For objectives (b)-(c), operation modes and adaptation levels need to be pre-specified for an off-chip implementation, while this is not required for an on-chip implementation. In contrast, objective (d) that addresses in field hardware-level failures can only be achieved by an on-chip implementation. Off-chip calibration is easier to implement compared to on-chip calibration since the latter requires the design of significant on-chip resources. However, an off-chip calibration is run on an ATE, while in an on-chip calibration the process takes place entirely and automatically on-chip. Finally, an off-chip calibration runs slower compared to an on-chip calibration due to the communication with the ATE.

Often the same calibration mechanism is used to achieve multiple of the above objectives. For example, the configuration setting for each operation mode may take into account process variations and non-idealities so as to achieve the most advantageous performance trade-off for each operation mode. In this case, the configuration settings are unique for each chip.

IV. Lock-less locking of programmable AMS ICs

A. Locking Principle

We argue that for programmable AMS ICs with off-chip calibration it is not required to insert additional circuitry on-chip, i.e., a lock, in order to introduce key-bits. Instead, we can take advantage of the embedded programmable fabric so as to naturally perform a lock-less locking operation, as shown in Fig. 2. Specifically, the configuration settings resulting from the calibration algorithm can be treated as secret keys or, equivalently, the programming bits can be treated as secret key-bits. An attacker who possesses the netlist can determine the secret keys if the calibration algorithm is known or can be

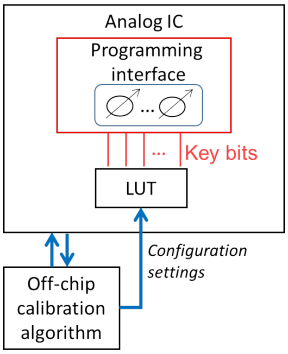


Fig. 2: Lock-less locking via the programmability interface.

determined. Thus, a requirement for the locking to be effective is that the calibration algorithm that produces the configuration settings is an asset of the design owner and complex enough to be determined by the attacker. The calibration algorithm is also kept secret and is not shared with any untrusted and potentially malicious party. Using invalid programming bits will result in untuning the circuit inciting complete loss of functionality or significant performance degradation, that is, one or more performances will lie far outside their allowable specification range.

B. On-chip calibration

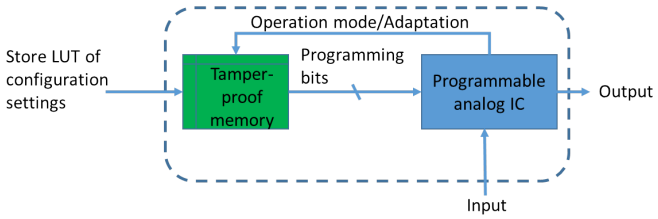
In the case of on-chip calibration, an attacker that extracts the netlist will also have at hand the hardware that implements the calibration feedback loop and, thereby, it may be fairly easy to extract the calibration algorithm. In this scenario, we can envision logic locking of the digital section of the calibration feedback loop [23], thus in this work we do not treat this scenario.

C. Key management

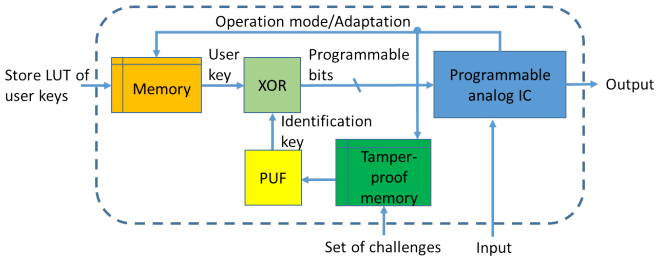
The possible secret key management schemes are the same ones used by any locking methodology, for digital or AMS ICs alike. One option is to directly store the LUT with the configuration settings into a TPM, as shown in Fig. 3a. A second option, illustrated in Fig. 3b, makes use of a PUF. The PUF needs to take at least as many challenges as the total number of configuration settings. This key management scheme uses a number of triplets {challenge, user key, configuration setting} that equals the number of configuration settings. The challenges are pre-stored in a TPM. For a target configuration setting, the corresponding challenge is driven to the PUF from the TPM and the PUF generates a secret identification key. The user key is defined such that when XORed with the identification key the target configuration setting is produced. In both schemes, for updating its operation mode or for an adaptation to the environment, the circuit commands dynamically the memories to load the corresponding programming bits.

D. Cost

Exploiting the embedded programmability for performing the locking operation presents significant advantages in terms



(a) Key management scheme based on directly storing the LUT of configuration settings into a TPM.



(b) Key management scheme using a PUF.

Fig. 3: Key management schemes.

of implementation cost. The key directly applies to the existing digitally-controlled tuning knobs without requiring the addition of any particular circuit for implementing a lock. The AMS IC is left completely intact and, thereby, there is no need for redesign, no extra design iterations, no changes in the design flow, and, most importantly, no performance degradation. In addition, the proposed approach does not increase the power or area of the AMS IC itself. The power and area overheads are only due to the key management scheme, which is the minimum unavoidable overhead for any IC locking technique for digital and AMS ICs alike. It should be noted also that in the context of a SoC the key management scheme is typically shared for enabling security for all other blocks of the SoC, thus it should not be considered as an overhead for a specific block. Thus, overall, the proposed locking approach offers the minimum possible implementation cost.

E. Testing/off-chip calibration phase

There are four options for performing the testing/off-chip calibration phase in such a way that: (a) resilience is achieved against remarking by gaining control over the number of activated functional chips; (b) the secrecy of the calibration algorithm and configuration settings is preserved. These options are:

1) The testing/off-chip calibration phase is performed in a trusted test facility.
2) The foundry returns the manufactured chips to the design house and owner of the IC, where the testing/off-chip calibration steps are performed in a secured environment.
3) Only structural defect-oriented testing is performed at the untrusted test facility, which does not require calibrating the chip first. Thereafter, calibration is performed by the trusted design house. Performing structural testing only is possible for designs with well-centered performances

that have an 100% parametric yield. A metric to assess the centering of a performance is its C_{pk} value, where $C_{pk} = \min\left[\frac{\text{USL}-\mu}{3\sigma}, \frac{\mu-\text{LSL}}{3\sigma}\right]$, USL and LSL denote the upper and lower specification limits of the performance, respectively, and μ and σ denote the mean and standard deviation of the performance, respectively. Structural test approaches that are generic or specific to an AMS IC class are continuously being proposed [47]–[51], and recently introduced industrial analog defect simulators [52], [53] assist in performing efficiently fault simulation towards test generation and test quality assessment.

4) The testing/calibration phase is performed in an untrusted test facility using secured remote calibration based on asymmetric cryptography [12]. More specifically, multiple test/calibration iterations are carried out for searching for the best configuration settings. In each step of the calibration algorithm, the next configuration setting is dictated by tests done using the current configuration setting. The trusted design house holds and runs the calibration algorithm and generates the next best configuration setting, while the untrusted test facility generates the test results for a given configuration setting. In each step, the design house communicates the configuration setting securely to the chip which is interfaced to the ATE in the untrusted test facility. The test result produced in the test facility is sent back to the trusted party to drive the generation of the next configuration setting, and so forth. If the calibration ends based on stop criteria and the performance specifications are not met, then the IC is labeled as faulty. To avoid remarking, the design house can deliberately load into the on-chip TPM largely offset configuration settings to render the chip totally malfunctional. For functional ICs, at the end of the calibration, the configuration settings that correctly activate the IC are stored into the on-chip TPM. Note that the calibration is automated on both sides and that the test results do not need to be secured.

F. Protection against IP/IC piracy threat models

1) Cloning: The proposed locking methodology can serve as a countermeasure against cloning by a malicious SoC integrator, foundry, or end-user that performs reverse-engineering. An attacker can extract the circuit architecture and netlist, but lacks the information on how the circuit is tuned. Thus, the circuit is rendered unclonable via protecting its tuning settings and the calibration algorithm that produces them.

2) Overbuilding and remarking: It serves also as a countermeasure against overbuilding by a malicious foundry and remarking by a malicious test facility since the design owner can take control over chip activation, as described in detail in Section IV-E.

3) Recycling: It can offer resilience against recycling only if the key management scheme in Fig. 3b is used and the unique user keys are re-loaded every time at power-on.

G. Security Analysis

We consider the most favorable threat model for an attacker. We assume that the attacker has full capabilities, i.e., has the circuit netlist and access to an unlocked oracle chip. Herein, we list the foreseen attacks, based also on all the known attacks in the literature in both the analog and digital domains, and we argue about the resilience offered by the proposed locking methodology.

1) Attacks in digital domain: Known attacks in the digital domain aiming at breaking logic locking techniques, such as the most lethal Boolean Satisfiability (SAT)-based attack [54], are not applicable. The reason is that an AMS IC has no Boolean representation and the input and/or output are analog. Such attacks also assume scan infrastructure or I/O access. Scan infrastructure is often not used inside digital cores of AMS ICs as AMS ICs are tested as a whole in the analog domain. Moreover, I/O access may not be granted in an AMS IC. The reason is that I/O nodes of AMS ICs may be high frequency nodes sensitive to parasitics and if not necessary they should not be routed to I/O pads since this would degrade the AMS IC performance.

2) Removal attacks: Removal attacks aim at removing or bypassing the lock. They are not applicable since the proposed locking methodology is lock-less; the key directly applies to existing tuning knobs into the design.

3) Attacks on biasing locking: The proposed locking methodology uses as key-bits the programming bits of tuning knobs, whereas a large class of tuning knobs are bias sources. All attacks on biasing locking work by considering the obfuscated component within the bias source [32]–[35]. In the proposed locking methodology, bias sources are not obfuscated; only their programming bits are kept secret. Therefore, attacks on biasing locking are not applicable.

4) Brute-force and multi-objective optimization attacks: The brute-force attack consists in applying random combinations of programming bits, i.e., keys, until a key is found that unlocks the circuit, i.e., brings all the performances within the acceptable specification range. Instead, the attacker can employ a multi-objective optimization algorithm, such as gradient descent, simulated annealing, Genetic Algorithm (GA), etc., to search more efficiently in the key space. More specifically, the attacker can formulate an optimization problem $\min_{key} \sum_j w_j |f_j(key) - s_j|$, where $f_j(key)$ is the function relating performance j with the key, s_j denotes the specification of performance j, and w_j are weight factors. The function $f_j(key)$ is intricate without a known closed-form relationship and is computed by invoking a circuit simulator. Such an optimization attack based on a GA is proposed in [34] and was originally used for breaking biasing locking. For AMS ICs it is likely that a number of keys result in a satisfactory performance trade-off, although this number is typically a very small fraction of all keys. For example, tuning knobs are typically used to generate bias currents or voltages that correctly set the DC operating point of the circuit. Keys that are "close" to the correct key will generate approximate biases that lead to correct functionality with a tolerated performance trade-off degradation. For every iteration of these attacks where a

specific key is evaluated, the circuit is simulated to compute the performances, and the performances are compared to the performances of the oracle chip.

Resilience against these attacks is proportional to the key size and to the simulation time. For AMS ICs one simulation run to compute all performances can be extremely time-consuming, thus the attacker in practice can afford carrying out just a few iterations. For our case study circuit, the key size is 194 bits and simulation time is in the order of one day. Thus, these attacks are infeasible.

To reduce the attack time, instead of simulating all the performances using appropriate test benches, the attacker could define a test stimulus and test duration and use as criterion some metric that assesses the similarity of the simulated response and the actual response from the oracle chip. Alternatively, the attacker could remove the entire key management scheme and re-fabricate the chip such that the key can be sourced directly as input and a brute-force or optimization attack can be performed entirely and faster in hardware. In both scenarios, the attack time is reduced, but the large key space still offers resilience. Furthermore, concerning the latter scenario, re-fabrication is clearly impractical for the attacker.

One workaround for the attacker could be dividing the circuit into sub-blocks, tracing key-bits to sub-blocks, and enabling smaller brute-force and multi-objective optimization attacks at sub-block level. This is not possible for two reasons. First, an AMS IC typically has internal feedback loops that involve multiple sub-blocks each, thus sub-blocks cannot be considered individually. Second, the performances of sub-blocks are not documented in the datasheet and the oracle does not offer access to sub-blocks for measurements.

With that said, there is another important defense against these attacks which is the fact that configuration settings are unique for each chip taking into account inter-die variations. These attacks become meaningful only if the extracted key from simulation of the nominal design can be used to set a good starting point for launching a gradient search for quickly calibrating any chip.

5) Revealing the calibration algorithm: The attacker may target speculating the calibration algorithm by studying the circuit architecture. However, very often the calibration algorithm is very specific to the design and customized by the designer. Thus, the attacker must have a very high and specialized expertise and a thorough understanding of the design so as to be able to conceive the underlying calibration algorithm. Revealing the calibration algorithm is a new type of attack specific to the countermeasure that is proposed. In general, it opens a discussion for securing and obfuscating calibration algorithms when they are considered to be a valuable asset of the design.

H. Applicability

The discussion in Section III remains quite general. In fact, the calibration scheme varies from one circuit class to another, and for a given circuit class it varies also from one architecture to another. Furthermore, for each architecture several calibration strategies exist.

Embedding programmability in AMS ICs becomes essential for pushing the performance limits of the technology, for course/fine tuning to compensate against process variability and non-idealities especially in advance technology nodes, for introducing multiple modes into the design, and for in-field calibration to deal with latent defects and aging and improve reliability and functional safety features. Programmability is also extensively used in digitally-assisted analog designs and digital centric AMS IC architectures [55], where the goal is to make a thoughtful shift of functionality from the analog into the digital domain, in order to alleviate analog design complexity, and then compensate low-precision with digital correction.

Several programmable versions exist for any AMS IC class, including filters, data converters, phase-locked loops (PLLs), RF transceivers, and mm-Wave circuits.

Programmability may vary from a few bits for calibrating single blocks up to tens or hundreds of bits for calibrating complete systems. For logic locking techniques, a necessary but not sufficient condition for thwarting counter-attacks is that the key size is 64 bits or higher. The reason is that the attacker can run fast simulations at a high abstraction-level, i.e., gate-level or register-transfer level. For AMS ICs, however, simulations need to be run at transistor-level since behavioral models are not available to the attacker. Even then, a behavioral model approximates functionality and cannot faithfully capture transistor-level effects, while it does not have sufficient detail to embed into it the lock. On the other hand, transistor-level simulation time can be very long, i.e., in the order of days for several AMS ICs such as data converters, PLLs, and RF transceivers, while in many cases a transistor-level simulation of the complete mixed analog-digital system is unfeasible and a divide-and-conquer approach has to be followed for pre-silicon verification. For this reason, for many AMS ICs the large key size condition can be safely relaxed to a few tens of bits.

In summary, the proposed locking methodology finds a wide applicability across several AMS IC classes and is in agreement with the current trend of embedding programmability into AMS ICs. Thanks also to the time-complexity of analog simulations, it may not necessarily require high programmability.

V. Case study

A. Programmable $\Sigma\Delta$ modulator

The recent increase in wireless communication standards has pushed the development of multi-standard RF transceivers [56]–[58], where the same RF transceiver circuit serves for establishing communication using several standards. Different standards have different requirements in terms of sensitivity, center frequency, bandwidth, resolution, etc. Therefore, it is necessary that many blocks within the RF transceiver are made programmable, in order to be able to adapt its specifications to the requirements imposed by the target standard. The configuration of the blocks is performed thanks to judiciously inserted tuning knobs which are controlled by digital programming bits.

Our case study for demonstrating the proposed locking methodology is a programmable band-pass RF $\Sigma\Delta$ modulator

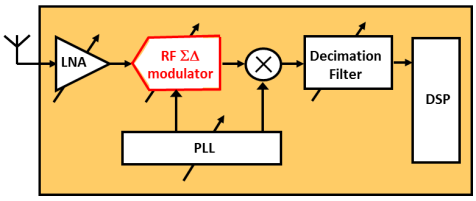


Fig. 4: Highly-digitized RF receiver architecture.

[59] designed in the context of a highly-digitized, multi-standard RF receiver, as illustrated in Fig. 4. The ΣΔ modulator is used to directly convert the RF signal at the output of the low noise amplifier (LNA) to the digital domain. It over-samples the analog input signal and generates a high-frequency, low-resolution 1-bit digital signal at its output. This signal is then down-converted by a digital mixer and filtered using a digital decimation filter.

The RF ΣΔ modulator is designed in a 65nm CMOS process. Its block-level schematic is illustrated in Fig. 5. At the higher level, it is composed of a band-pass loop filter, a comparator, feedback DACs, a tunable delay block, and an output buffer. It is designed specifically for an RF receiver that establishes communication using several standards within the frequency range from 1.7GHz to 2.8GHz, such as Bluetooth, ZigBee, and WiFi 802.11b, as well as several standards dedicated to second generation 2G and fourth generation 4G broadband cellular networks, such as GSM, GPRS, EDGE, LTE1800, LTE2100, and LTE2600. All the sub-blocks of the ΣΔ modulator are made programmable to set its operation mode according to the target standard and at the same time to tune its performance trade-off on a per-chip basis in the presence of process variations. In total, the ΣΔ modulator uses a 194-bit programming word. The partitioning of the 194 programming bits into the different sub-blocks and their utility is as follows:

- The center frequency of the loop filter is tuned in the presence of process variations. For this purpose, in each LC tank inside the loop filter there are two capacitor arrays, namely an array C_c with a 6-bit configuration word for coarse-tuning and another array C_f with a 6-bit configuration word for fine-tuning.
- The quality factor of each LC tank is tuned in the presence of process variations using a 10-bit configuration word that controls the current of a negative transconductance $-Gm$.
- A 8-bit configuration word is dedicated to trimming the biasing current of the tunable delay block so as to adapt it to the sampling frequency F_s of the modulator.
- The 56-bit configuration word dedicated to the comparator, the 7-bit configuration word dedicated to each Gm inside the loop filter, and the 7-bit configuration word dedicated to each DAC in the feedback loop are independent of the center frequency tuning and are used to trim the biasing currents so as to compensate for process variations.
- The biasing current of the output buffer is controlled through an 8-bit configuration word with the objective of adapting the 1-bit output of the modulator to its off-chip load.

B. Calibration Interface

During calibration, the programming bits $\{b_1, b_2, \cdots, b_n\}$ of the configuration word are streamed into the chip propagating through a serial shift register composed of cascaded D-FFs, as illustrated in Fig. 6. Once the full configuration word is loaded into the serial register, it is latched to a parallel register where it is stored, activating the ΣΔ modulator. The parallel register is refreshed in every iteration of the calibration algorithm. In the field of operation, the configuration words resulting from calibration are pre-stored in the memory, and the configuration word corresponding to the desired operation mode is loaded.

The tuning knobs in the design are of two types, namely programmable binary-weighted capacitor arrays inside the LC tanks of the loop filter and programmable binary-weighted current mirrors setting the biases in all other blocks, i.e., transconductances Gm and negative transconductances $-Gm$ inside the loop filter, comparator, feedback DACs, tunable delay block, and output buffer.

Fig. 7 depicts a programmable binary-weighted current mirror. The programming bits control which mirroring branches are turned on, contributing to the adjustment of the mirroring ratio. The resultant bias current is given by $\sum_{i=1}^{n} b_i \cdot 2^i \cdot I_{ref}$. Fig. 8 depicts a programmable capacitor array. In this case, the programming bits control which capacitors contribute to the equivalent capacitance, which is given by $\sum_{i=1}^{n} b_i \cdot 2^i \cdot C$.

The fact that the tuning knobs are binary-weighted has also a security implication: no two different configuration words can produce the same tuning knob values, i.e., bias currents or capacitances. This limits the number of keys that can establish an acceptable performance trade-off. Only the keys that produce tuning knob values in the neighborhood of the nominal tuning knob values can meet this objective, and this number of keys is a tiny fraction of the key space.

C. Calibration algorithm

An off-chip calibration algorithm is used to tune a fabricated chip for a given standard in the presence of process variations. Thus, the configuration settings per standard are unique for each fabricated chip. The calibration algorithm is as follows:

1) The comparator is configured as a buffer by deactivating its driving clock.
2) The input signal is disabled by turning off the input transconductance Gm_1.
3) The feedback DACs current is turned off.
4) The third LC tank in the loop filter is put in oscillation mode by setting its Q-enhancement transconductance $-Gm$ to its maximum.
5) The capacitor arrays C_c and C_f of the third LC tank in the loop filter are tuned until the output frequency is equal to the desired center frequency dictated by the standard.
6) The Q-enhancement transconductance $-Gm$ of the third LC tank in the loop filter is reduced gradually until oscillation vanishes.

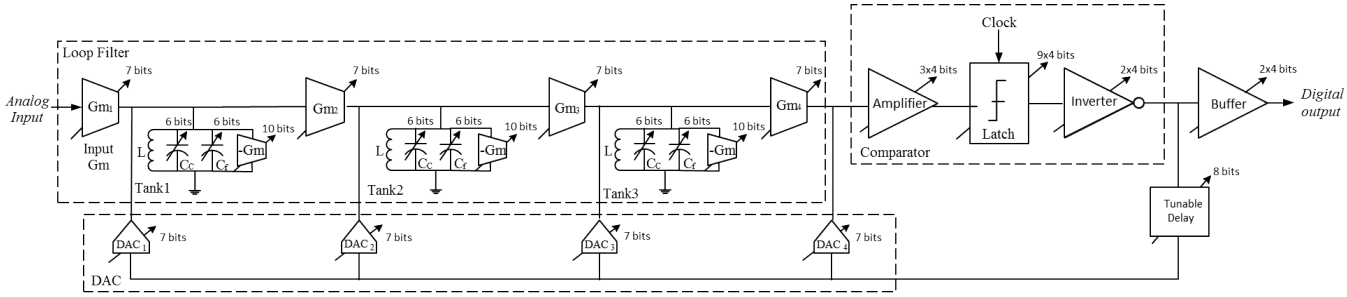


Fig. 5: Architecture of $\Sigma\Delta$ modulator.

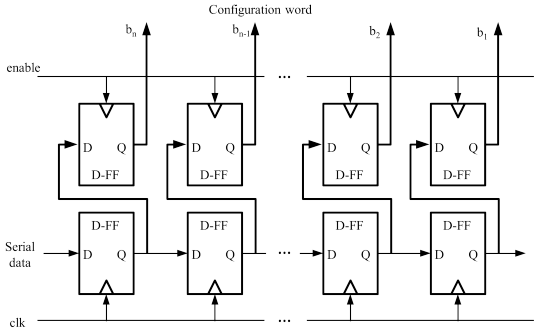


Fig. 6: Configuration word loading and storage.

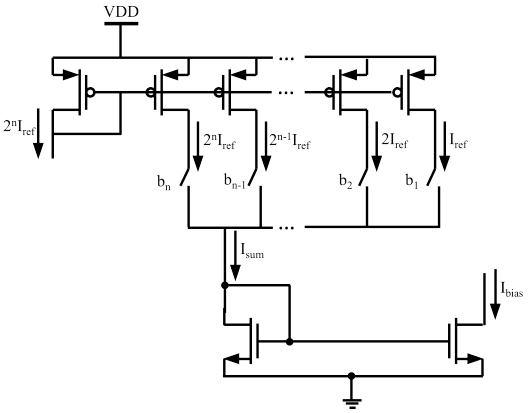


Fig. 7: Programmable current mirror.

7) Steps 1-6 are repeated for the second and first LC tanks in the loop filter.
8) The feedback loop is restored.
9) The $\Sigma\Delta$ modulator is put in the operating mode by applying an RF input signal with frequency F_0.
10) The sampling frequency is set to $F_s = 4 \cdot F_0$.
11) The tunable delay is set according to F_s.
12) The tuning knobs of the input transconductance Gm_1, feedback DACs, and comparator are initialized to their nominal values determined by simulation.
13) An iterative procedure is used to determine the optimal configuration words of these blocks in the presence of process variations through the improvement of the measured Signal-to-Noise Ratio (SNR) and Spurious Free Dynamic Range (SFDR) of the $\Sigma\Delta$ modulator.

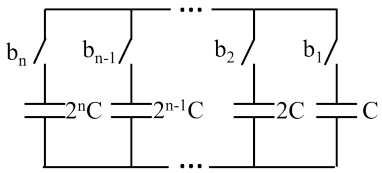


Fig. 8: Programmable capacitor array.

VI. Experimental Results

A. Locking efficiency

Our experiment is conducted using hardware measurements on the actual fabricated chip. Without loss of generality, we consider the operation mode with the maximum center frequency, e.g., 2.8GHz.

The circuit has several performances, including SNR, SFDR, input third-order intercept point (IIP3), etc., and locking succeeds when at least one performance violates its specification for any incorrect key. Ideally, for an incorrect key all performances should violate their specifications, and the further they lie away from their specifications the more efficient the locking is.

The locking efficiency is first assessed by applying random keys and comparing the resultant SNR with the SNR of the unlocked circuit when the correct key is applied. For this measurement, we consider an input sinusoidal signal with frequency 2.8GHz and power -14dBm. Fig. 9 shows the SNR across 5000 randomly generated keys and the correct key. As it can be seen, the correct key stands out resulting in an SNR of over 60dB, while for incorrect keys the SNR is less than 30dB. The average SNR across incorrect keys is around 10dB, while the maximum and minimum observed SNR is 26.6dB and -21.1dB, respectively. This experiment shows that applying incorrect keys degrades drastically the SNR performance, thus the locking objective is achieved.

Next, we present measurement results for the rest of the performances considering three keys, namely the correct key and the "best" and "worst" incorrect keys in Fig. 9, i.e., the incorrect keys resulting in the highest and lower observed SNR, respectively.

Fig. 10 shows the Power Spectral Density (PSD) at the output of the $\Sigma\Delta$ modulator. As it can be seen, for the "worst" incorrect key there is no noise shaping shown by the "V" shape

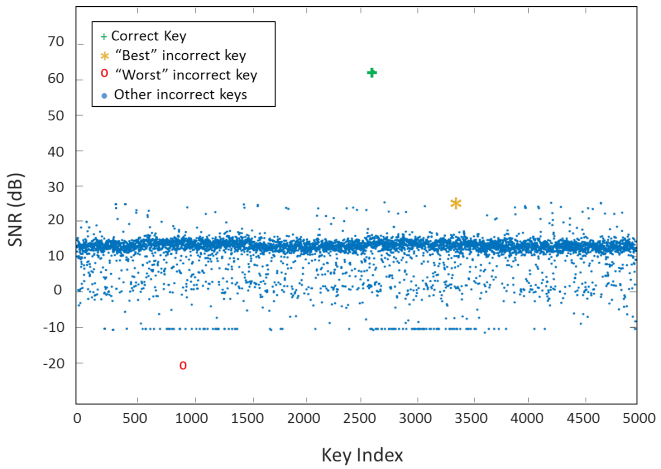


Fig. 9: SNR measured for correct and incorrect keys.

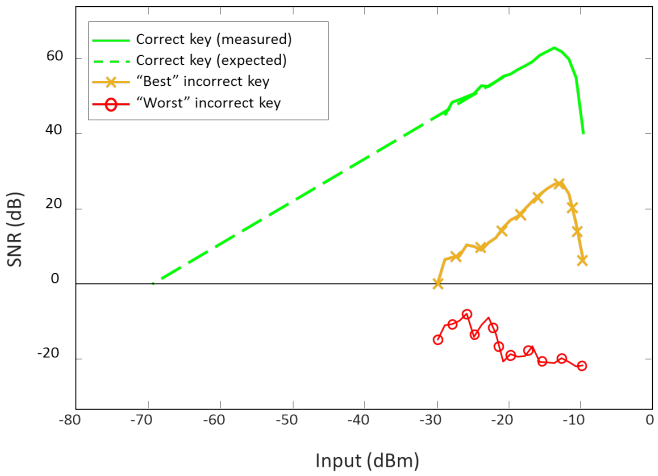


Fig. 11: SNR versus input power for the correct key and the "best" and "worst" incorrect keys in Fig. 9.

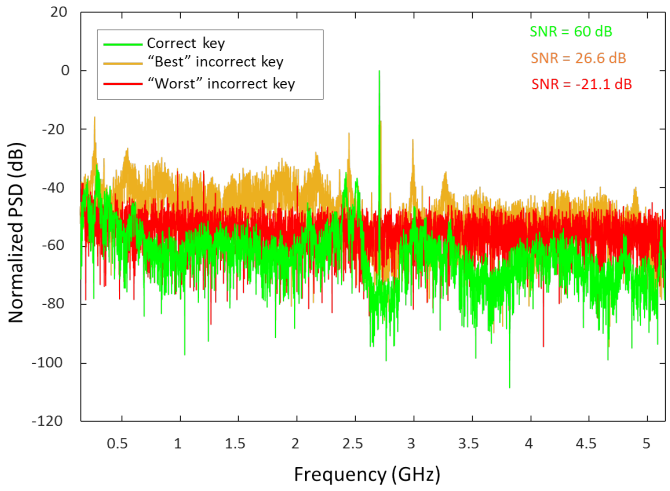


Fig. 10: PSD measured for the correct key and the "best" and "worst" incorrect keys in Fig. 9.

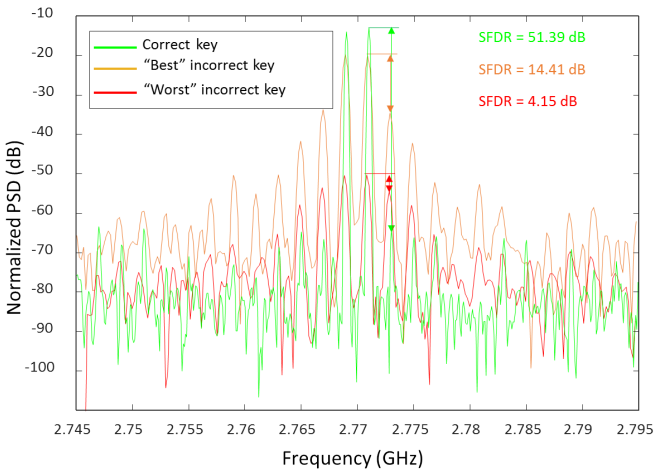


Fig. 12: SFDR versus input power for the correct key and the "best" and "worst" incorrect keys in Fig. 9.

in the band-of-interest, which is the main characteristic of a band-pass $\Sigma\Delta$ modulator. The signal is permanently buried under the noise level. For the "best" incorrect key, noise shaping is observed; however, there are some harmonics in the band-of-interest reducing the SNR to 26.6dB.

Fig. 11 plots the dynamic range. While Fig. 9 shows the SNR for an input power of -14dBm corresponding to the maximum obtained SNR, Fig. 11 plots the SNR for different input power values with a step of 1dBm. As it can be seen, the "best" incorrect key results in significantly reduced dynamic range, while for the "worst" incorrect key the curve does not show the expected knee behavior and is permanently under 0dB.

Fig. 12 plots the SFDR measured by applying a two-tone input, where the two tones have the same power and a frequency difference of 10MHz. SFDR is the difference between the power of the fundamental and the third harmonic. As it can be seen, the nominal SFDR is 51.39dB, whereas the "best" and "worst" incorrect keys result in significantly reduced SFDR values of 14.41dB and 4.15dB, respectively.

Fig. 13 shows the output fundamental power and the third-order intermodulation (IM_3) product versus the input power, from which the IIP3 can be determined. We apply a two-tone input, where the two tones have the same input power and a frequency difference of 2MHz. The output fundamental power and IM_3 are measured by sweeping the input power from -13dB to -17dB with a step size of 1dB. As it can be seen, the unlocked circuit has a nominal IIP3 of 8dBm, while the "best" and "worst" incorrect keys result in a significantly reduced IIP3 of -8dBm and -12dBm, respectively.

As a final note, the same experiment was repeated for other center frequencies in the band 1.7-2.8GHz and other fabricated chip samples and qualitatively the results were identical.

B. Resilience to attacks

1) Brute-force and multi-objective optimization attacks: The key space is 2^{194}, thus the search space for brute-force and multi-objective optimization attacks is huge. Besides, these attacks are implemented at simulation-level, and one

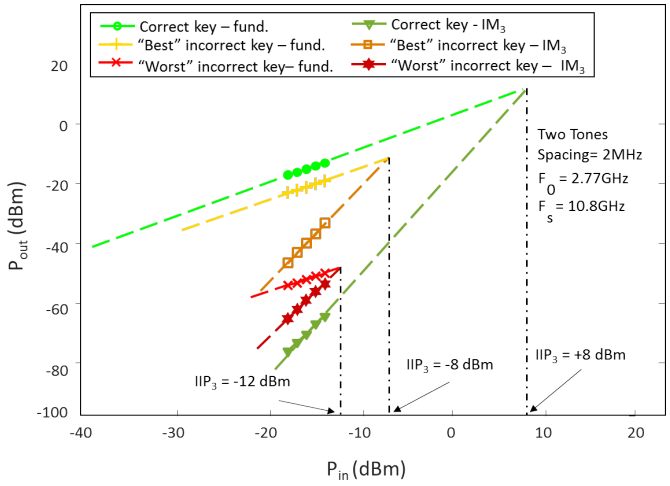


Fig. 13: IIP3 for the correct key and the "best" and "worst" incorrect keys in Fig. 9.

transistor-level simulation to compute the circuit performances is extremely time-consuming. More specifically, for a single key and an 8192-point Fast Fourier Transform (FFT), it takes about 30 minutes to simulate the SNR for a given input power, 4.5 hours to simulate the SNR across the complete input range, 45 minutes to simulate the SFDR, and 3.75 hours to simulate the IIP3. The overall simulation time per iteration will be in the order of hours, thus only a few iterations can be performed in practice and it will be impossible to "hit" a good performance trade-off within an affordable simulation time.

2) Revealing the calibration algorithm: There are many aspects in the algorithm that make it very complex, thus hindering the attacker's ability to recover it. It is design-specific and its steps cannot be easily retraced by conjecture even under the assumption that the attacker has strong AMS IC design expertise. In particular: (a) The circuit needs to be reconfigured into appropriate test benches multiple times during calibration; (b) The order with which the different sub-blocks should be calibrated is very specific; (c) Most sub-blocks are included in a feedback loop which prohibits calibrating sub-blocks individually, given also that the target performances of individual blocks per standard are unknown to the attacker; (d) The calibration of many sub-blocks requires initial programming bits that are dictated by design simulation and are unknown to the attacker. If other than these programming bits are used then convergence in a reasonable time is not guaranteed.

VII. Conclusions

We proposed a locking methodology for the class of programmable AMS ICs that leverages the existing programmability based on tuning knobs. No lock needs to be inserted into the design since the existing tuning knobs take the role of sub-locks. The key is the concatenation of the programming bits of tuning knobs. The owner of the IC needs to keep secret the configuration settings per standard and per chip, as well as the calibration algorithm that is used to generate these configuration settings. The locking methodology presents many advantages that allow its wide adoption by AMS IC designers, i.e., it is strictly non-intrusive to the design, it incurs zero power and area overheads except for the overheads due to the key management scheme that are common to any locking methodology, and it neither requires any change in the AMS IC design flow nor any design re-iterations. We demonstrated the locking methodology on a $\Sigma\Delta$ ADC that has a 194-bit programmability so as to be part of a multi-standard RF receiver. Hardware measurements showed very efficient locking as any invalid key resulted in dramatically degraded performance trade-offs.

References

[1] M. Rostami, F. Koushanfar, and R. Karri, "A primer on hardware security: Models, methods, and metrics," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1283–1295, 2014.

[2] R. Torrance and D. James, "The state-of-the-art in semiconductor reverse engineering," in *Proc. IEEE/ACM Design Automation Conference*, 2011, pp. 333–338.

[3] B. Lippmann, M. Werner, N. Unverricht, A. Singla, P. Egger, A. Dübotzky, H. Gieser, M. Rasche, O. Kellermann, and H. Graeb, "Integrated flow for reverse engineering of nanoscale technologies," in *Proc. Asia and South Pacific Design Automation Conference*, 2019, p. 82–89.

[4] U. Guin, K. Huang, D. DiMase, J. M. Carulli, M. Tehranipoor, and Y. Makris, "Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1207–1228, 2014.

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

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

[7] J. Rajendran, M. Sam, O. Sinanoglu, and R. Karri, "Security analysis of integrated circuit camouflaging," in *Proc. ACM Conference on Computer and Communications Security*, 2013, pp. 709–720.

[8] 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 Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 64 – 77, 2017.

[9] 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 Transactions on Design Automation of Electronic Systems*, vol. 24, no. 6, pp. 65:1–65:36, 2019.

[10] A. Chakraborty, N. G. Jayasankaran, Y. Liu, J. Rajendran, O. Sinanoglu, A. Srivastava, Y. Xie, M. Yasin, and M. Zuzak, "Keynote: A disquisition on logic locking," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 10, pp. 1952–1972, 2020.

[11] M. Yasin, S. M. Saeed, J. Rajendran, and O. Sinanoglu, "Activation of logic encrypted chips: pre-test or post-test?," in *Proc. Design, Automation & Test in Europe Conference & Exhibition*, 2016, pp. 139–144.

[12] J. A. Roy, F. Koushanfar, and I. L. Markov, "Ending piracy of integrated circuits," *IEEE Computer*, vol. 43, no. 10, pp. 30–38, 2010.

[13] C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: A tutorial," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1126–1141, 2014.

[14] I. Polian, "Security Aspects of Analog and Mixed-Signal Circuits," in *Proc. IEEE International Mixed-Signal Testing Workshop*, 2016.

[15] A. Antonopoulos, C. Kapatsori, and Y. Makris, "Trusted analog/mixed-signal/RF ICs: A survey and a perspective," *IEEE Design & Test*, vol. 34, no. 6, pp. 63–76, 2017.

[16] M. M. Alam, S. Chowdhury, B. Park, D. Munzer, N. Maghari, M. Tehranipoor, and D. Forte, "Challenges and Opportunities in Analog and Mixed Signal (AMS) Integrated Circuit (IC) Security," *Journal of Hardware and Systems Security*, vol. 2, no. 1, pp. 15–32, 2018.

[17] A. Sanabria-Borbón, N. G. Jayasankaran, J. Hu, J. Rajendran, and E. Sánchez-Sinencio, "Analog/RF IP protection: Attack models, defense techniques, and challenges," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 1, pp. 36–41, 2021.

[18] D. H. K. Hoe, J. Rajendran, and R. Karri, "Towards secure analog designs: A secure sense amplifier using memristors," in *Proc. IEEE Computer Society Annual Symposium on VLSI*, 2014.

[19] V. V. Rao and I. Savidis, "Protecting analog circuits with parameter biasing obfuscation," in *Proc. IEEE Latin American Test Symposium*, 2017.

[20] J. Wang, C. Shi, A. Sanabria-Borbon, E. Sánchez-Sinencio, and J. Hu, "Thwarting analog IC piracy via combinational locking," in *Proc. IEEE International Test Conference*, 2017.

[21] V. V. Rao and I. Savidis, "Mesh based obfuscation of analog circuit properties," in *IEEE International Symposium on Circuits and Systems*, 2019.

[22] G. Volanis, Y. Lu, S. Govinda, R. Nimmalapudi, A. Antonopoulos, A. Marshall, and Y. Makris, "Analog performance locking through neural network-based biasing," in *Proc. IEEE VLSI Test Symposium*, 2019.

[23] N. G. Jayasankaran, A. Sanabria Borbon, E. Sanchez Sinencio, J. Hu, and J. Rajendran, "Towards provably-secure analog and mixed-signal locking against overproduction," *IEEE Transactions on Emerging Topics in Computing*, 2020.

[24] S. G. Rao Nimmalapudi, G. Volanis, Y. Lu, A. Antonopoulos, A. Marshall, and Y. Makris, "Range-controlled floating-gate transistors: A unified solution for unlocking and calibrating analog ICs," in *Proc. Design, Automation & Test in Europe Conference*, 2020.

[25] J. Leonhard, M. Yasin, S. Turk, M. Nabeel, M.-M. Louërat, R. Chotin-Avot, H. Aboushady, O. Sinanoglu, and H.-G. Stratigopoulos, "MixLock: Securing mixed-signal circuits via logic locking," in *Proc. Design, Automation & Test in Europe Conference*, 2019.

[26] J. Leonhard, M.-M. Louërat, H. Aboushady, O. Sinanoglu, and H.-G. Stratigopoulos, "Mixed-signal hardware security using MixLock: Demonstration in an audio application," in *International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design*, 2019.

[27] J. Leonhard, N. Limaye, S. Turk, A. Sayed, A. R. Rizo Díaz, H. Aboushady, O. Sinanoglu, and H.-G. Stratigopoulos, "Digitally-assisted mixed-signal circuit security," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2021.

[28] Y. Bi, J. S. Yuan, and Y. Jin, "Beyond the interconnections: split manufacturing in RF designs," *Electronics*, vol. 4, no. 3, pp. 541–564, 2015.

[29] A. Ash-Saki and S. Ghosh, "How multi-threshold designs can protect analog IPs," in *Proc. IEEE International Conference on Computer Design*, 2018, pp. 464–471.

[30] J. Leonhard, A. Sayed, M.-M. Louërat, H. Aboushady, and H.-G. Stratigopoulos, "Analog and mixed-signal IC security via sizing camouflaging," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 5, pp. 822–835, 2021.

[31] M. Elshamy, A. Sayed, M.-M. Louërat, A. Rhouni, H. Aboushady, and H.-G. Stratigopoulos, "Securing programmable analog ICs against piracy," in *Proc. Design, Automation & Test in Europe Conference*, 2020.

[32] N. G. Jayasankaran, A. Sanabria-Borbón, A. Abuellil, E. Sánchez-Sinencio, J. Hu, and J. Rajendran, "Breaking analog locking techniques," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 10, pp. 2157–2170, 2020.

[33] V. V. Rao, K. Juretus, and I. Savidis, "Security vulnerabilities of obfuscated analog circuits," in *IEEE International Symposium on Circuits and Systems*, 2020.

[34] R. Y. Acharya, S. Chowdhury, F. Ganji, and D. Forte, "Attack of the genes: Finding keys and parameters of locked analog ICs using genetic algorithm," in *IEEE International Symposium on Hardware Oriented Security and Trust*, 2020, pp. 284–294.

[35] J. Leonhard, M. Elshamy, M.-M. Louërat, and H.-G. Stratigopoulos, "Breaking analog biasing locking techniques via re-synthesis," in *Proceedings of the 26th Asia and South Pacific Design Automation Conference*, 2021, p. 555–560.

[36] K. Juretus, V. Venugopal Rao, and I. Savidis, "Securing analog mixed-signal integrated circuits through shared dependencies," in *Proc. ACM Great Lakes Symposium on VLSI*, 2019.

[37] V. Natarajan, S. Sen, A. Banerjee, A. Chatterjee, G. Srinivasan, and F. Taenzler, "Analog signature-driven postmanufacture multidimensional tuning of RF systems," *IEEE Design & Test of Computers*, vol. 27, no. 6, pp. 6–17, 2010.

[38] J. W. Jeong, A. Nassery, J. N. Kitchen, and S. Ozev, "Built-in self-test and digital calibration of zero-IF RF transceivers," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 6, pp. 2286–2298, 2016.

[39] Y. Lu, K. S. Subramani, H. Huang, N. Kupp, K. Huang, and Y. Makris, "A comparative study of one-shot statistical calibration methods for analog/RF ICs," in *Proc. IEEE International Test Conference*, 2015, Paper 21.3.

[40] M. Andraud, H.-G. Stratigopoulos, and E. Simeu, "One-shot non-intrusive calibration against process variations for analog/RF circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 11, pp. 2022–2035, 2016.

[41] F. Cilici, M. J. Barragan, S. Mir, E. Lauga-Larroze, S. Bourdel, and G. Leger, "Yield recovery of mm-wave power amplifiers using variable decoupling cells and one-shot statistical calibration," in *IEEE International Symposium on Circuits and Systems*, 2019.

[42] C. Maxey, G. Creech, S. Raman, J. Rockway, K. Groves, T. Quach, L. Orlando, and A. Mattamana, "Mixed-signal SoCs with in situ self-healing circuitry," *IEEE Design & Test of Computers*, vol. 29, no. 6, pp. 27–39, 2012.

[43] S.M. Bowers, K. Sengupta, B.D. Parker, and A. Hajimiri, "Integrated self-healing for mm-wave power amplifiers," *IEEE Transactions on Microwave Theory and Techniques*, vol. 61, no. 3, pp. 352–363, 2013.

[44] S. Sen, V. Natarajan, S. Devarakond, and A. Chatterjee, "Process-variation tolerant channel-adaptive virtully zero-margin low-power wireless receiver systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 12, pp. 1764–1777, 2014.

[45] D. Banerjee, S. K. Devarakond, X. Wang, S. Sen, and A. Chatterjee, "Real-time use-aware adaptive RF transceiver systems for energy efficiency under BER constraints," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 8, pp. 1209–1222, 2015.

[46] S. Lee, C. Shi, J. Wang, A. Sanabria, H. Osman, J. Hu, and E. Sánchez-Sinencio, "A built-in self-test and *In Situ* analog circuit optimization platform," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 10, pp. 3445–3458, 2018.

[47] G. Huertas, D. Vázquez, E. J. Peralías, A. Rueda, and J. L. Huertas, "Testing mixed-signal cores: A practical oscillation-based test in an analog macrocell," *IEEE Design & Test of Computers*, vol. 19, no. 6, pp. 73–82, 2002.

[48] L. Abdallah, H.-G. Stratigopoulos, S. Mir, and J. Altet, "Defect-oriented non-intrusive RF test using on-chip temperature sensors," in *Proc. IEEE VLSI Test Symposium*, 2013.

[49] 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, the VLSI Journal*, vol. 55, pp. 393–400, 2016.

[50] M. Ince, E. Yilmaz, W. Fu, J. Park, K. Nagaraj, L. Winemberg, and S. Ozev, "Digital built-in self-test for phased locked loops to enable fault detection," in *IEEE European Test Symposium*, 2019.

[51] 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 Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 6, pp. 2580–2593, 2021.

[52] S. Sunter, K. Jurga, and A. Laidler, "Using mixed-signal defect simulation to close the loop between design and test," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 12, pp. 2313–2322, 2016.

[53] V. Zivkovic and A. Schaldenbrand, "Requirements for industrial analog fault-simulator," in *International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design*, 2019, pp. 61–64.

[54] P. Subramanyan, S. Ray, and S. Malik, "Evaluating the security of logic encryption algorithms," in *Proc. IEEE International Symposium on Hardware Oriented Security and Trust*, 2015.

[55] B. Murmann, "Digitally assisted analog circuits," *IEEE Micro*, vol. 26, no. 2, pp. 38–47, 2006.

[56] M. Ingels, V. Giannini, J. Borremans, G. Mandal, B. Debaillie, P. Van Wesemael, T. Sano, T. Yamamoto, D. Hauspie, J. Van Driessche, and J. Craninckx, "A 5 mm^2 40 nm LP CMOS transceiver for a software-defined radio platform," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 12, pp. 2794–2805, 2010.

[57] S. Li, J. Li, X. Gu, H. Wang, C. Li, J. Wu, and M. Tang, "Reconfigurable All-Band RF CMOS Transceiver for GPS/GLONASS/Galileo/Beidou With Digitally Assisted Calibration," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 9, pp. 1814–1827, 2015.

[58] D. Haghighitalab, D. Belfort, A. Kilic, A. Benlarbi-Delai, and H. Aboushady, "A 2.4 GHz ISM-band highly digitized receiver based on a variable gain LNA and a subsampled Sigma-Delta ADC," *Analog Integrated Circuits and Signal Processing*, vol. 95, no. 2, pp. 259–270, 2018.

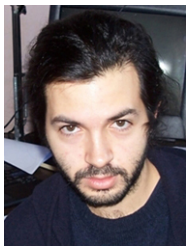
[59] A. Sayed, T. Badran, M. Louërat, and H. Aboushady, "A 1.5-to-3.0GHz tunable RF sigma-delta ADC with a fixed set of coefficients and a programmable loop delay," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 9, pp. 1559–1563, 2020.

Mohamed Elshamy received the M.Sc. degree in Electronics and Communications Engineering from Cairo University, Giza, Egypt, in 2015 and the Ph.D degree from Sorbonne Université, Paris, France, in 2021. He is currently a research assistant in the laboratory of nanotechnology application in electronics at the Electronics Research Institute, Cairo, Egypt. His research interests include hardware security, analog and mixed-signal circuits, nano-electronics, and memristors.

Alhassan Sayed received the B.Sc. and the M.Sc. degrees in Electrical Engineering, from the Electronics and Communications Department, Minia University, Minia, Egypt, in 2007 and 2010, respectively. He obtained his Ph.D. degree in Electrical Engineering and Computer Science from Sorbonne University, Paris, France, in 2016. He also spent 2 years (2017-2019) in a postdoctoral research position at the same University. Dr. Sayed is currently an Assistant Professor at Minia University, Minia, Egypt. His research interests include Sigma-Delta modulation, analog and RF circuit design, Analog-to-Digital conversion, and low noise amplifiers.

Marie-Minerve Louërat received the M.Sc. degree in Electrical Engineering and the Ph.D. degree from Université Paris Sud, Orsay, France, in 1983 and 1986 respectively. In 1986 she joined the Centre National de la Recherche Scientifique (CNRS), France. She started at Fluids, Automation and Thermal Systems Laboratory, Université Paris Sud-CNRS, while teaching electronics. In 1992, she moved to the Computer Science Laboratory (LIP6), University Pierre et Marie Curie (now Sorbonne Université)-CNRS, France, while teaching VLSI. Between 2013 and 2018, she was the head of the System on Chip Department at LIP6. Dr. Louërat's research interest is electronic design automation methods and tools for analogue and mixed-signal circuits and systems. Most of her research activities have been supported by contracts, through academic and industrial cooperative projects in the framework of the FP7, Eureka/MEDEA, Catrene, Penta, and H2020 Projects. She published papers on static timing analysis, analogue and AMS design automation, analogue-to-digital converters, AMS system modelling and simulation, and test and security of AMS circuits and systems. She is a member of the AMS Working Group of Accellera Systems Initiative and contributed to standardize the AMS extension of SystemC since 2010. She has served on the Technical Program Committee of Design, Automation, and Test in European Conference (DATE) and several others international conferences. She co-chaired the Free Silicon Conference (FSIC) in 2019, Paris, France.

Hassan Aboushady (Senior Member, IEEE) received the B.Sc. degree in Electrical Engineering from Cairo University, Egypt, in 1993, the M.Sc. and Ph.D. degrees in Electrical Engineering and Computer Science from Sorbonne University, Paris, France, in 1996 and 2002 respectively. Dr. Aboushady is currently an Associate Professor at Sorbonne University. His research interests include Sigma-Delta modulation, Analog/RF circuit design, Analog-to-Digital and Digital-to-Analog conversion, as well as Security in analog and mixed-signal circuits. He is the author and co-author of more than 70 publications in these areas. He is the recipient of the 2004 best paper award in the IEEE Design Automation and Test in Europe Conference, as well as the recipient and the co-recipient of the 2nd and the 3rd best student paper awards of the IEEE Midwest Symposium on Circuits and Systems in 2000 and 2003 respectively. Dr. Aboushady is an IEEE-CAS distinguished lecturer and a member of the IEEE Circuits and Systems for Communications Committee (CASCOM). He also served as an Associate Editor of the IEEE Transactions on Circuits and Systems II: Express Briefs.



Haralampos-G. Stratigopoulos (Member, IEEE) received the Diploma in electrical and computer engineering from the National Technical University of Athens, Athens, Greece, in 2001 and the Ph.D. in electrical engineering from Yale University, New Haven, USA, in 2006. He is a Research Director with the French National Center for Scientific Research (CNRS) at LIP6 Laboratory, Sorbonne Université, Paris, France. His main research interests are in the areas of design-for-test for analog, mixed-signal, RF circuits and systems, machine learning, hardware security, and neuromorphic computing. He was the General Chair of the 2015 IEEE International Mixed-Signal Testing Workshop (IMSTW) and the Program Chair of the 2017 IEEE European Test Symposium (ETS). He has served on the Technical Program Committees of Design, Automation, and Test in Europe Conference (DATE), Design Automation Conference (DAC), IEEE International Conference on Computer-Aided Design (ICCAD), IEEE European Test Symposium (ETS), IEEE International Test Conference (ITC), IEEE VLSI Test Symposium (VTS), and several others international conferences. He has served as an Associate Editor of IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on Circuits and Systems I: Regular Papers, IEEE Design & Test, and Springer Journal of Electronic Testing: Theory & Applications. He received the Best Paper Award in the 2009, 2012, and 2015 IEEE European Test Symposium (ETS).