Analog and Mixed-Signal IC Security Via Sizing Camouflaging

Julian Leonhard, Alhassan Sayed, Marie-Minerve Louërat, Hassan Aboushady, Haralampos-G. Stratigopoulos

Analog and Mixed-Signal IC Security Via Sizing Camouflaging

Julian Leonhard, *Student Member, IEEE*, Alhassan Sayed, Marie-Minerve Louërat, Hassan Aboushady, *Member, IEEE*, and Haralampos-G. Stratigopoulos, *Member, IEEE*

***Abstract*—We treat the problem of analog Integrated Circuit (IC) obfuscation towards Intellectual Property (IP) protection against reverse engineering. Obfuscation is achieved by camouflaging the effective geometry of layout components via the use of fake contacts, which originally were proposed for gate camouflaging in digital ICs. We present a library of obfuscated layout components, we give recommendations for effective camouflaging, we discuss foreseen attacks and the achieved resiliency, and we propose security metrics for assessing the hardness of reverse engineering. The proposed methodology is demonstrated on an operational amplifier and an RF $\Sigma\Delta$ Analog-to-Digital Converter (ADC).**

***Index Terms*—Hardware security and trust, analog and mixed-signal integrated circuits, IP/IC piracy, reverse engineering, design obfuscation, camouflaging.**

I. Introduction

Reverse engineering has become one of the major hardware security threats. It can be used to reconstruct the netlist of an Integrated Circuit (IC) and infer its functionality. Nowadays, instrumentation and software tools are broadly available to successfully reverse engineer any IC regardless of the technology node [1], [2]. While in most countries reverse engineering is legal for checking ICs for piracy or patent infringements, there are many ways it can be misused. For example, a company may gather intelligence so as to reduce its competitive disadvantage against the IC owner company. A reverse engineered IC can be cloned and sold as original without licensing, thus resulting in revenue and know-how losses for the IC owner. Reverse engineering can be used also to locate the root-of-trust and security primitives of an IC and gather information for launching a successful attack that leaks secret data. Finally, it can be used with the aim to judiciously insert a Hardware Trojan which when triggered can degrade or disable functionality.

Defense strategies against reverse engineering include locking and physical design obfuscation. Locking aims at inserting a lock into the design such that unless the valid key is used the functionality breaks. Physical design obfuscation, on the other hand, aims at making "stealthy" alterations in a design using mechanisms at the device and interconnect level, resulting in an extracted netlist that is "deceiving" for the attacker.

Extensive reviews of existing defense strategies for digital ICs are provided in [3]–[8]. For analog ICs, hardware security techniques in general lag seriously behind compared to their digital counterparts and the solution space is largely unexplored [9]–[11].

A well-known physical obfuscation mechanism is based on fake contacts between metal layers and poly, diffusion or metal layers[1] [12]. True contacts span the entire dielectric to connect the two layers, whereas fake contacts have a thin gap creating an open-circuit. Fake contacts are 100% CMOS compatible requiring no foundry process changes [12]. An attacker cannot differentiate between true and fake contacts as they appear identical under a microscope and by slicing the die it will be unlikely to pass through the thin gap. Besides, fake contacts are distributed at different heights and this would require slicing the die in several pieces which is infeasible. Another approach is to make true contacts with magnesium (Mg), which displays very good electrical conductivity, and fake contacts with magnesium oxide (MgO), which is a perfect insulator [13]. When delayering a protected IC the Mg contacts oxidize within minutes to MgO, thereby destroying the information where real and where fake contacts are placed in the layout. A remedy for the attacker could be to delayer in an oxygen-free environment, an approach that would make costs and efforts soar prohibitively. Generally, fake contacts can be leveraged to inconspicuously blend extra circuitry into the IC which, however, is inactive and completely irrelevant for the functionality of the IC. In [14], fake contacts are used to design a camouflaged cell that can perform either as an XOR, NAND, or NOR gate according to which contacts are true and fake. The designer can replace some standard gate cells with this camouflaged cell to obfuscate the functionality.

In this paper, we propose an analog IC camouflaging technique based on the use of fake contacts. Compared to gate camouflaging, the proposed analog IC camouflaging works differently. Gate camouflaging hides the gate functionality, whereas analog IC camouflaging hides the correct sizing of the components, such that the extracted netlist from the reverse-engineered circuit has deceiving sizing and, thereby, unacceptable performance trade-off. Gate camouflaging requires camouflaging a large percentage of gates so as to increase the reverse engineering hardness, which inevitably results in

J. Leonhard is with Sorbonne Université, CNRS, LIP6, Paris, France (e-mail: Julian.Leonhard@lip6.fr).

A. Sayed is with Sorbonne Université, CNRS, LIP6, Paris, France and Minia University, Minia, Egypt (e-mail: Alhassan.Sayed@lip6.fr).

M.-M. Louërat is with Sorbonne Université, CNRS, LIP6, Paris, France (e-mail: Marie-Minerve.Louerat@lip6.fr).

H. Aboushady is with Sorbonne Université, CNRS, LIP6, Paris, France (e-mail: Hassan.Aboushady@lip6.fr).

H.-G. Stratigopoulos is with Sorbonne Université, CNRS, LIP6, Paris, France (e-mail: Haralampos.Stratigopoulos@lip6.fr).

[1]For the sake of simplicity but without loss of generality we will refer to all types of interconnects including vias as contacts.

large area, delay, and power overheads [15]. In contrast, in analog IC camouflaging it suffices to obfuscate a small number of components, thus the overheads can be well-controlled and can be practically negligible. In gate camouflaging the attacker can recognize the camouflaged gates, which can be informative for launching attacks, whereas in analog IC camouflaging the attacker will have to consider every component as potentially obfuscated, which increases dramatically the hardness of reverse engineering. We present a library of obfuscated analog layout components that is sufficient for camouflaging virtually any analog IC and we provide recommendations to designers for best camouflaging practices. We also discuss foreseen attacks and the resiliency offered by the proposed technique. Finally, we propose security metrics specific to analog ICs to quantify the hardness of reverse engineering. The technique is demonstrated on two case studies, namely a Miller operational amplifier (op-amp) and an RF $\Sigma\Delta$ Analog-to-Digital Converter (ADC).

The rest of the paper is structured as follows. In Section II, we discuss the prior art in analog IC locking and physical design obfuscation. In Section III, we provide an overview of the analog IC camouflaging technique. In Section IV, we present the library of obfuscated analog layout components. In Section V, we provide recommendations for best camouflaging practices. In Section VI, we discuss foreseen attacks and the achieved resiliency. In Section VII, we develop security metrics for quantifying the hardness of reverse engineering. In Section VIII, we present our experimental results on the chosen two case studies. Section IX concludes the paper.

II. Prior art in analog IC locking and obfuscation

For analog ICs the vast majority of published works focus on locking. The challenge is to insert a lock that is driven by tens of key-bits, e.g. 64 bits for high resilience against a trial and error brute-force attack, yet without inducing any performance penalty.

The majority of published works on locking consider inserting a lock into the biasing circuitry of the analog IC, such that unless the correct key is applied the analog IC is incorrectly biased, thus resulting in one or more performances residing outside the specification limits. In [16], it is proposed to lock the body-biasing of a matched transistor pair with a lock mechanism that is based on a memristor crossbar, where the key-bits control the programming of the memristors. In [17], it is proposed to replace the transistors used to set the biasing with parallel-connected transistors. The key-bits control which of these transistors are "on", where the aggregate width of the "on" transistors equals the width of the original obfuscated transistor. In [18], it is shown how to lock current mirrors by adding extra branches that are controlled by the key-bits. The resultant biasing current depends on which branches are "on", as well as on the dimensions of the mirroring transistors in these branches. In [19], it is proposed to add a neural network on-chip that receives as input an analog key in the form of DC biases and produces at its output the desired biases. The neural network is trained to implement a delta function at the correct key, that is, any invalid key will produce incorrect biases. In [20], an attack is proposed based on Satisfiability Modulo Theory (SMT) that can break all biasing locking techniques.

Another category of approaches consider the calibration mechanism inserted into analog ICs. In [21], it is proposed to apply logic locking to the digital optimizer in the feedback loop that maps on-chip performance measurements to an appropriate tuning knob setting that compensates for process variations. In [22], it is assumed that the analog IC has embedded analog floating-gate transistors (AFGTs) for fine-tuning of the performances. A lock mechanism is proposed that limits the programmability range of AFGTs when an invalid key is applied, thus disabling the capacity for correct calibration. A different approach is taken in [23], where it is proposed to lock highly-digitized analog ICs naturally via their programmability fabric. In this case, the configuration settings that simultaneously compensate for process variations and set the operation mode are kept secret. The calibration algorithm that produces these configuration settings must be complex enough to hinder the attacker from reverse engineering it.

A third approach considers locking analog ICs by applying logic locking to their digital section that is in the signal path [24]. A demonstrator of this technique was presented in [25], where the ADC in an audio signal processing chain is locked. One can listen to the effect of locking which translates into glitches in the output audio signal.

Furthermore, it is also possible to consider compound techniques where more than one locking mechanisms are embedded simultaneously. For example, in [26], the analog section of a mixed-signal circuit is locked with biasing transistor obfuscation and its digital section is secured with logic locking.

Finally, there is only one existing technique in the literature for physical design obfuscation which leverages multi-threshold voltage (V_{th}) transistors often used in analog ICs [27]. More specifically, it is proposed to replace few normal-V_{th} transistors with low-V_{th} or high-V_{th} and re-design the circuit so as to meet the intent specifications. The underlying assumption is that an attacker that reverse-engineers the chip and extracts the netlist cannot distinguish the type of the transistor. This approach can be viewed as transistor type camouflaging. For a circuit with n transistors, an upper bound for the search space size will be 3^n, after excluding the transistors whose type is unambiguous and counting a pair of transistors that must have the same type, i.e. matched transistors in a differential pair or current mirroring transistors, as one. While re-designing requires some effort, the designers should be willing to undertake this effort, in order to protect the intellectual property of their design.

III. Analog IC camouflaging

A. Threat model

The proposed analog IC camouflaging is a defense against reverse-engineering attempted by a malicious end-user. In our threat model, the attacker legally purchases a functional chip from the market. We assume that the attacker has access to the technology Process Design Kit (PDK) and has full capabilities to reverse-engineer the chip and resolve geometries down to

sub-gate-level sizes, thus recovering an exact schematic and layout. The attacker can also purchase a second chip that can be used as an oracle, i.e., for applying inputs and observing the outputs.

The proposed analog IC camouflaging does not protect an Intellectual Property (IP) block from a malicious System-on-Chip (SoC) integrator or a malicious foundry that fabricates the IC, since the IP/IC owner inevitably shares with these potentially untrusted parties the blueprint of the IP/IC, e.g., GDS-II file, whereby fake contacts are directly revealed.

B. Sizing camouflaging

The proposed analog IC camouflaging consists in inconspicuously hiding by means of fake contacts the active geometry of layout components and, thereby, the actual sizing of schematic components extracted from reverse-engineering. The methodology takes advantage of the special handcrafted layout techniques used in analog designs for improving component matching, tolerating process variations, and achieving compact layouts [28].

In particular, non-minimum size transistors are most often laid out as several sub-transistors connected in parallel and sharing diffusion strips, known as gate fingers. Common-centroid layouts are also preferred for transistor pairs that are required to be well-matched. Similarly, resistors are laid out in a serpentine serial connection of unit resistors and capacitors are laid out as capacitor banks consisting of several unit capacitors.

The underlying idea is to use fake contacts so as to add seemingly connected yet in reality inactive and electrically disabled gate fingers, unit resistors, and unit capacitors. In this way, the nominal sizing of components, i.e. the effective width of transistors and the values of resistors and capacitors, is camouflaged.

In Section IV, we will present in detail camouflaged layout versions of analog components using fake contacts. These camouflaged layout versions can be parametrized into PCells to compose a library of camouflaged PCells that is combined with the library of standard PCells and is seamlessly integrated into the design flow. A camouflaged PCell combines the functionality of the standard PCell while also adding extra electrically disabled instances. Building the library of camouflaged PCell is a one-time effort for each technology node and thereafter can be reused for readily obfuscating any design. Moreover, the same design principle can be reused for every technology node. For a target component to be resized, the designer will simply have to replace the standard PCell with the camouflaged PCell and set the parameters of the camouflaged PCell. This set of parameters includes the active sizing, as well as the number of extra inactive instances and their locations, i.e. the arrangement of active and inactive instances.

C. The defender perspective: design flows with camouflaging

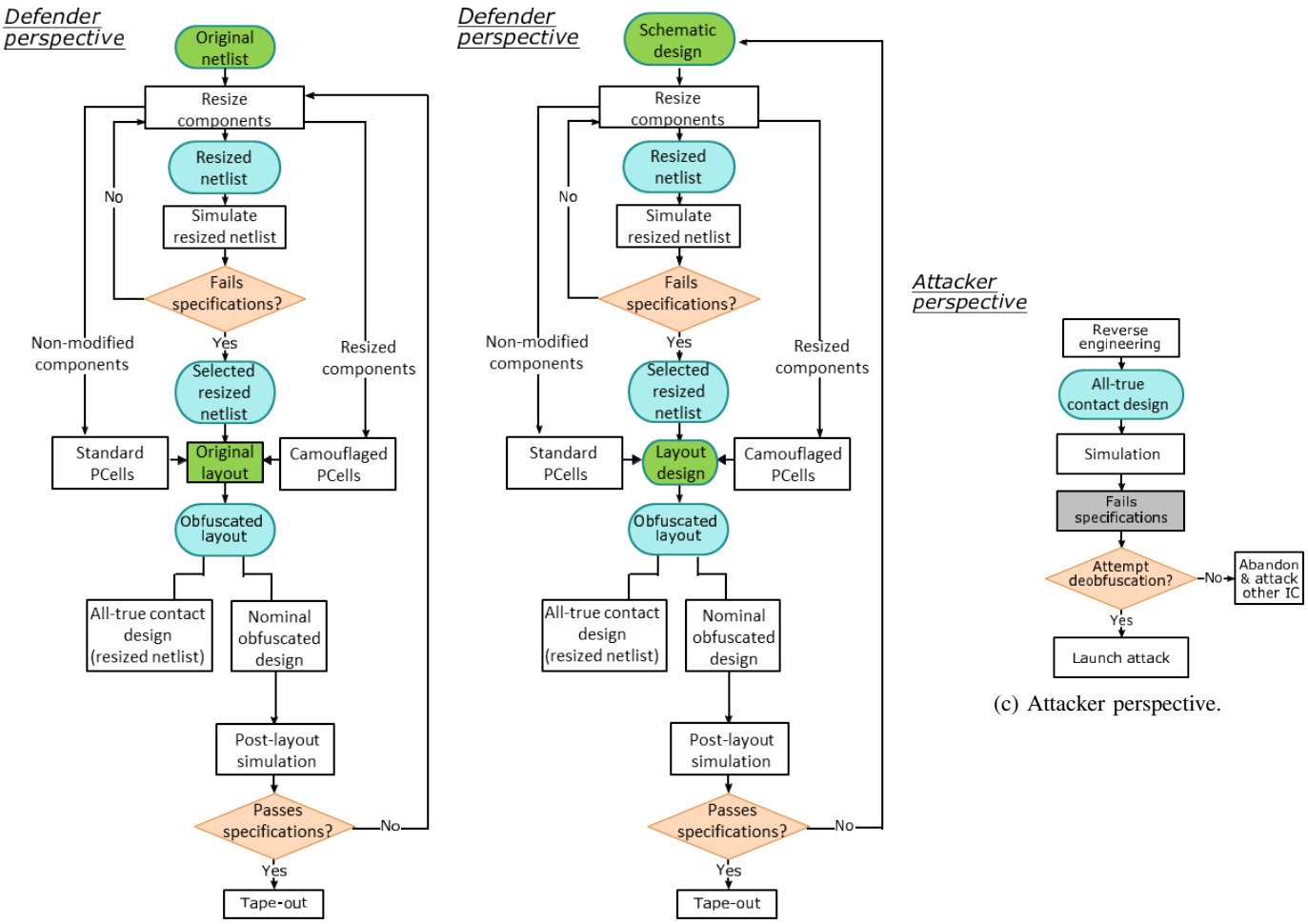
We can distinguish two design flows, namely camouflaging of an existing design, shown in Fig. 1a, and involving camouflaging already from the design phase, shown in Fig. 1b.

1) Camouflaging an existing design: The defender has the *original* design, including the original netlist and layout, which we refer to as the *nominal non-obfuscated* design. Beginning with the original netlist, the defender will perform re-design iterations, shown with the inner loop in Fig. 1a, where in each step a set of components is resized and the resized netlist is simulated to obtain the performances. This inner loop stops when a suitable resized netlist is found that has one or more performances failing their specifications.

With this selected resized netlist, the defender will next obfuscate the original layout. The layout of non-modified components remains unchanged, while the layout of resized components is replaced with an obfuscated layout version using the library of camouflaged PCells, as explained in Section III-B. This replacement possibly will require changes in the floor-planning and routing, in order to fit into the original layout the camouflaged layout versions of the resized components. The resulting layout is an obfuscated layout that is electrically equivalent to the original layout since the resizing is cancelled out by the use of fake contacts. Therefore, the obfuscated layout embeds the nominal design which we refer to as the *nominal obfuscated* design. However, if fake contacts cannot be distinguished from true contacts and are all reckoned as true, then the obfuscated layout can be deceivingly thought to embed the resized netlist which we refer to as the *all-true contact* design.

Compared to an original component layout, a camouflaged component layout will add extra parasitics which, albeit small, may perturb the intent performance trade-off of the original design. Perturbation may result also from changes in the floor-planning and routing. To ensure that the nominal obfuscated design does not incur any performance penalty with respect to the nominal non-obfuscated design, as a final step, the defender will perform post-layout simulation to evaluate the performances of the nominal obfuscated design. If unacceptable performance degradation is noticed, then the defender will have to repeat the camouflaging procedure, as illustrated by the outer loop in Fig. 1a. The defender can identify the modified components that are the root-cause of this degradation and will target resizing another set of components that results in no degradation. With this outer loop, obfuscation via sizing camouflaging can be viewed as an additional step in the design flow that can be performed on top of the original design.

2) Involving camouflaging in the design phase: The designer knows in advance before actually starting the design that the design should be protected against reverse-engineering. In this scenario, camouflaging is fully integrated into the design flow. The designer will proceed as normal and will first design the circuit at schematic-level with no camouflaging in mind. Once the intent design specifications are met at schematic-level and before moving to layout design, the designer will perform the resizing operation for camouflaging, shown with the inner loop in Fig. 1b, similarly to the design flow in Fig. 1a. Thereafter, the layout will be designed as normal using camouflaged layout versions for the resized components. Thus, in this case, floor-planning and routing naturally takes into consideration the camouflaged layout versions of components. Once the layout is completed, post-layout simulations will be



(a) Defender perspective when camouflaging an existing design.

(b) Defender perspective when involving camouflaging in the design phase.

(c) Attacker perspective.

Fig. 1: Overview of analog IC camouflaging.

performed as normal. Typically, several design iterations take place until post-layout performances are satisfactory, as shown with the outer loop in Fig. 1b. During this design optimization, the designer will change the nominal component values, i.e. transistor dimensions, etc., will perform changes in the layout, floor-planning, and routing, and may also perform topology modifications. This outer loop is not related to the obfuscation objective. However, for every iteration of the outer loop, we may have to repeat the inner loop which is related to the obfuscation objective.

D. The defender perspective: objectives

The defender has the following main two objectives:

1) For the design flow in Fig. 1a, maximize the performance penalty of the all-true contact design with respect to the nominal non-obfuscated design. For the design flow in Fig. 1b, maximize the performance penalty of the all-true contact design with respect to the specified performance trade-off.

2) For the design flow in Fig. 1a, minimize any performance penalty of the nominal obfuscated design with respect to the nominal non-obfuscated design. For the design flow in 1b, the nominal obfuscated design should meet the specified performance trade-off.

We can define two additional objectives:

3) Minimize the obfuscation area overhead.

4) Minimize the obfuscation design effort towards satisfying faster the above objectives 1 and 2.

For the design flow in Fig. 1a, minimizing the obfuscation design effort implies: (a) reducing the number of iterations of the inner loop and (b) reducing the number of iterations of the outer loop which, in turn, will reduce the number of the repetitions of the inner loop. As mentioned in Section III-C, the outer loop aims at correcting any performance penalty of the nominal obfuscated design with respect to the nominal non-obfuscated design. This performance penalty is due to camouflaged layout-induced parasitics and changes in the floor-planning and routing. Reducing this performance penalty will reduce the number of iterations of the outer loop and possibly may eliminate completely the need to enter into this loop, thus iterating over the inner loop only once.

For the design flow in Fig. 1b, the outer loop aims at design optimization such that post-layout performances meet the intent specifications. As mentioned in Section III-C, this outer loop is not related to the obfuscation objective, yet the inner loop which is related to this objective is revisited at every iteration of the outer loop. Therefore, for the design

flow in Fig. 1b, minimizing the obfuscation design effort implies: (a) reducing the number of iterations of the inner loop and (b) avoiding repeating the inner loop during outer loop iterations. The latter can be achieved by aiming at minimizing the effect of camouflaged layout-induced parasitics on post-layout performances. In this way, camouflaged layout-induced parasitics will not be among the root-causes of unsatisfactory post-layout performances which is what enables the outer loop. The set of resized components for obfuscation as well as their resizing values can be kept fixed during outer loop iterations. As long as the resized netlist, e.g. the all-true contact design, fails the specifications, it will not be necessary to repeat the inner loop and find another set of components to resize.

Therefore, minimizing the obfuscation design effort boils down to the following objectives:

4a) For both design flows, reduce the number of iterations of the inner loops in Figs. 1a and 1b towards satisfying faster objective 1.

4b) For both design flows, minimize camouflaged layout-induced parasitics towards satisfying faster objective 2.

4c) For the design flow in Fig. 1a, additionally minimize changes in the floor-planning and routing towards satisfying faster objective 2.

Recommendations for best camouflaging practices will be given in Section V.

E. The attacker perspective

Fig. 1c illustrates the attacker perspective. The attacker will initially perceive all contacts as true and only after running simulations will realize that the performances of the all-true contact design are not in agreement with those promised in the datasheet having a degraded performance trade-off with one or more specifications lying outside their specification range. At that point the attacker will understand that the design is obfuscated, but cannot tell which are the fake contacts and for that reason cannot tell which are the obfuscated components either. Every component is potentially an obfuscated one. As a result, the attacker will have extracted the architecture and netlist, but will not recover the sized netlist nor a correct layout and is hindered from replicating the functionality and performances promised in the datasheet. The attacker may choose to attack another unprotected IC promising similar functionality, or may decide to attempt an attack to de-obfuscate. Foreseen attacks will be detailed in Section VI and security metrics to assess the hardness of de-obfuscation will be given in Section VII.

IV. Library of camouflaged layout components

Herein, we provide a library of obfuscated layout versions of components that are most commonly met in analog layouts, including multiple gate-finger transistors, common-centroid layout of transistors, interdigitized transistors, serpentine resistors, and capacitor banks. Of course, this is a non-exhaustive list of possible obfuscated layout versions of such components, and a non-exhaustive list of components that can be obfuscated, i.e., it excludes inductors and diodes, but it largely suffices to camouflage the sizing of virtually any analog IC.

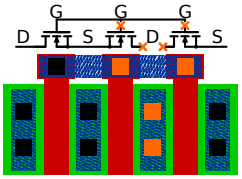


Fig. 2: Obfuscated multiple gate-finger transistor layout with its schematic. Diffusion, poly-silicon, and metal are drawn respectively in green, red, and blue. True contacts are drawn in black and fake contacts in orange.

1) Transistors: Multiple gate-finger transistors are parallel transistors of equal gate width where each transistor shares its inner diffusion regions for drain or source with its two neighbouring transistors. Fig. 2 shows an example of a compact transistor layout with 3 gate fingers. The inner diffusion regions control the state of 2 gate fingers at once, while the outer regions control a single finger. A transistor can be obfuscated by connecting extra gate fingers and deactivating them by using fake contacts in the drain or source terminals such that these nodes are floating. In Fig. 2, the fake contacts are shown with orange color, whereas true contacts are shown with black color. Two fake contacts are used to deactivate two gate fingers. Two fake contacts are also used to disconnect completely the two adjacent gates. This is preferred so as to reduce the parasitic load, but is only possible if no shared poly-silicon gate is drawn. The equivalent schematic with the open-circuits resulting from fake contacts is also shown on top of Fig. 2. In this example, the transistor has 1 active gate finger, but the attacker observes a transistor with a gate width 3 times larger.

Certain transistor arrangements, i.e., differential transistor pairs or current mirrors, require special layout techniques to ensure matching. Common-centroid layouts are typically used for differential transistor pairs ensuring that gradients across the die will impact both transistors equally. Fig. 3 shows a layout of a common-centroid differential transistor pair A and B showing an AXXBBXXA pattern, with X representing deactivated transistors due to the inserted fake contacts. The equivalent schematic is shown on top of Fig. 3. With the inserted fake contacts the attacker observes that A and B consist of 4 active transistors each while in reality they consist of 2. By changing the gate connections in Fig. 3 we can turn the circuit into an interdigitized current mirror with obfuscated current ratio between A and B according to where the fake contacts are placed. The actual current ratio will be invisible to the attacker.

2) Capacitors: The capacitor value of a capacitor bank can be obfuscated by adding extra capacitor units and disconnecting them through the use of fake contacts. Fig. 4 shows the side-view of an exemplary layout of a metal-insulator-metal (MIM) capacitor bank consisting of 2 parallel-connected unit capacitors. Metcap[2] and metal 2 (M2) are the respective plates of a capacitor. Through the use of fake contacts, shown with a thin gap, the right-hand capacitor is disconnected from

[2]Metcap is an additional layer used to realize MIM capacitors. Across different technologies this layer may be called differently.

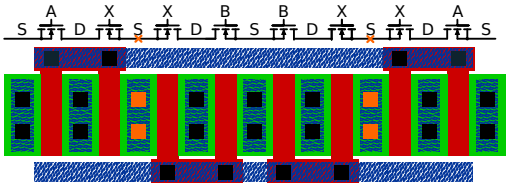


Fig. 3: Obfuscated common-centroid layout and schematic with AXXBBXXA pattern, where the letters A,B and X over the gates mark to which transistor structure the transistor layout below belongs to. X marks deactivated instances due to fake contacts. To not impair visibility the connections between respective sources and drains of A and B are not drawn.

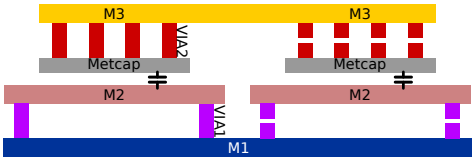


Fig. 4: Side-view of obfuscated capacitor bank layout. The obfuscated capacitor on the right has fake contacts seemingly connecting both its capacitor plates.

the capacitor bank. Both plates are disconnected so as to reduce parasitic capacitance to a minimum. In this example, the attacker observes an incorrect, two times bigger capacitor value.

3) Resistors: The value of a serpentine resistor can be obfuscated by adding extra unit resistors. As an example, Fig. 5 shows a serpentine resistor composed of 5 unit resistors. The idea is to use wiring across each unit resistor to create short-circuits and place fake contacts to cut the short-circuits for those unit resistors that will be active. Interestingly, in contrast to transistors and capacitors, fake contacts here are used to activate instances. In this example, the nominal resistance is $3R$, whereas the attacker observes a resistor value $k \cdot R$, but does not know k which could take any value in $\{0, \cdots, 5\}$.

A camouflaged PCell is readily built from the standard PCell and can be instantiated to implement any degree of resizing and any arrangement of active and inactive instances. It can be viewed as a standard PCell with a subset of contacts replaced with fake contacts, in order to deactivate the corresponding instances. The camouflaged PCell takes as parameters the standard PCell parameters, as well as the number and location of inactive instances. For example, for a camouflaged PCell of a multiple gate-finger transistor, the designer will have to set the nominal transistor dimensions, i.e. length, width, and number of gate fingers, the number of inactive extra gate fingers, as well as their arrangement with respect to the active gate fingers.

V. Recommendations for Analog IC Camouflaging

The number of components to resize, the degree of resizing per obfuscated component, and the selection of components to resize are driven by the objectives defined in Section III-D.

A. Number of components to resize and degree of resizing

With the proposed camouflaging approach, in the reverse-engineered netlist all components are potentially obfuscated

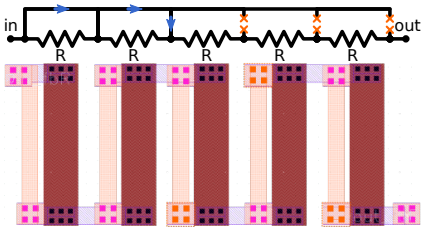


Fig. 5: Obfuscated serpentine resistor layout. Resistive poly, metal 1, metal 2, true contacts, fake contacts, and metal1-metal2 vias are shown respectively in dark red, blue, light orange, black squares, orange squares, and pink squares.

in the eye of the attacker. Therefore, the hardness of reverse-engineering does not depend on the number of resized components. We can turn this fact to our advantage and target resizing only a small number of components that is sufficient for achieving an all-true contact design that has a degraded performance trade-off (objective 1).

Achieving objective 1 is an easy task since analog ICs are very sensitive to component sizing. Although analog IC design optimization and centering can be a very time-consuming and tedious task requiring high expertise, here the defender aims at the "inverse" task, i.e. untuning the circuit and destroying the performance trade-off, which arguably can be achieved in an effortless way. It is not surprising if objective 1 is achieved by resizing a single component. In general, the first inner loops in the design flows in Figs. 1a and 1b should take only a few iterations to achieve objective 1 (objective 4a).

By only resizing a small number of components, we can meet additional objectives defined in Section III-D. Specifically: (a) obfuscation area overhead is kept at a minimum (objective 3); (b) total camouflaged layout-induced parasitics will be effectively minimized (objective 4b); and (c) for the design flow in Fig. 1a, minor changes in the floor-planing and routing will be required (objective 4c).

Note that objective 1 can also be met by distributing the resizing across many components and applying a smaller degree of resizing for each component. However, this strategy intuitively will be more time-consuming for meeting objective 1, requiring more iterations of the inner loops in Fig. 1a and 1b. Besides, in this way, the camouflaged layout-induced parasitics get distributed too and it will be more difficult controlling them. Moreover, it is not guaranteed that this strategy will overall reduce the obfuscation area overhead, and for the design flow in Fig. 1a it is likely that changes in the floor-planing and routing would be more significant. For these reasons, we recommend obfuscating a small number of components with the resizing required to satisfy objectives 1 and 2, and only when the resizing turns out to be very large try to distribute the resizing across more components. This last recommendation aims at avoiding having unnaturally large layout components that from the attacker perspective will look suspicious and likely obfuscated.

B. Degree of performance degradation

One question that arises is to what degree to degrade the performance trade-off of the all-true contact design. If the all-

true contact design is functional, showing small performance deviation outside the allowable specification range, then the cloned design can still be used in applications where the performance requirements are less stringent. Therefore, the defender goal should be to introduce a performance penalty in the all-true contact design at least to a point where it becomes of low-quality and unusable and, thereby, not appealing any more for cloning.

C. Selection of components to resize

The following guidelines can be used:

1) Selecting to resize components that largely influence the performance trade-off will result in a smaller number of resized components. This helps meeting several objectives as explained in Section V-A (i.e. objectives 1, 3, and 4). However, we recommend that the selection process should not follow any formal or established methodology, such as a sensitivity analysis, which ranks the components according to their influence. The reason is that the attacker may think of employing the exact same methodology to trace back the resized components. We argue that the best approach towards increasing the reverse engineering hardness is to randomly select components to obfuscate based on intuition about their influence.

2) On top of resizing a small number of components, avoiding resizing components that are connected to sensitive or high-frequency nodes will further minimize the effect of camouflaged layout-induced parasitics on the performance trade-off (objective 4b).

3) In Section IV, we presented obfuscation layout versions of transistors, resistors, and capacitors. This library can be extended to include other components, i.e. inductors. Clearly, adding extra inactive fingers to transistors results in much lower area overhead compared to adding extra inactive unit capacitors, unit resistors, or extending the coil of a wire inductor. Thus, priority should be given to obfuscating transistors rather than passive components towards low obfuscation area overhead (objective 3). In addition, for the design flow in Fig. 1a, this will reduce the required changes in the floor-planing and routing (objective 4c).

4) Regarding the design flow in Fig. 1a, selecting to resize components that have enough empty space in their periphery on the layout, i.e. they are located in layout areas that are not compact, will reduce the obfuscation area overhead (objective 3) and will avoid introducing changes in the layout that may require reexamining the floor-planning and routing (objective 4c). If components can be resized without changing the placement of surrounding components in the layout, then obfuscation area overhead will be zero. In general, in analog layouts many areas are left unoccupied, in order to leave sufficient space between sensitive blocks with the goal to mitigate electromagnetic interference, crosstalk, thermal-related issues, etc. This gives us large flexibility for inserting the camouflaged layout versions in the existing floor-planning. Since the resized portion of the component is seemingly connected with fake contacts making it inactive and electrically disabled, it should not change the profile of the circuit. In any case, minimum distances between adjacent objects as defined in the PDK should be respected and electromagnetic compatibility compliance should not be compromised.

5) If the to-be-protected circuit is a complex system consisting of a number of sub-blocks, then the straightforward approach would be to obfuscate every sub-block, i.e. resize components in every sub-block. However, this is not strictly necessary as the aim of obfuscation is to act on the global system-level performances. In other words, for complex systems it suffices to resize a small number of components in a few sub-blocks to obtain an all-true contact design with degraded performance (objective 1), thus also minimizing the obfuscation design effort (objective 4). We will discuss this case also in relation to foreseen attacks in Section VI.

6) A common layout practice found in analog layouts is the placement of dummy components for better matching properties and compensation of process variations. Existing dummy components can be seemingly connected to their neighboring active components if they have the same geometry via the use of fake contacts, thus naturally extending the resizing. In this way, we can naturally degrade further the performance trade-off of the all-true contact design (objective 1), reduce the obfuscation area overhead (objective 3), and iterate less over the inner loops in Figs. 1a and 1b (objective 4a). For the design flow in Fig. 1a, this additionally helps minimizing changes in the floor-planning and routing (objective 4c). However, this strategy should be followed conservatively and cautiously so as to maintain low camouflaged layout-induced parasitics.

VI. Attacks against analog IC camouflaging

1) Attacks on gate camouflaging: SAT-based attacks [29], [30] that have compromised the security of gate camouflaging techniques for digital ICs do not apply to analog ICs. The reason is that SAT solvers rely on Boolean algebra while analog circuits carry continuous-time signals.

2) Brute-force attack: The attacker will massively try different combinations of component sizing in the hope of eventually guessing a sizing that results in a satisfactory performance trade-off. Our defense is that the attacker is obliged to consider every component in the circuit as potentially obfuscated. The search space size is $\prod_{i=1}^{D} N_i$, where D is the number of components and N_i denotes the number of instances in the i-th component. This search space can be reduced if the attacker makes some informed assumptions, as it will be explained in more detail in Section VII. A second defense is the fact that analog simulation can be very time-consuming. Thus, in practice a very small fraction of the search space can be explored.

3) SMT-based attack: The SMT-based attack proposed in [20] can be used to speed up de-obfuscation as long as circuit equations can be written. In particular, for component i we can write an equation $y_i = \phi(\mathbf{q}^i)$, where $\mathbf{q}^i = [q_1^i, \cdots, q_{N_i}^i]$ is a string of key-bits of size N_i, N_i is the number of instances, and $q_j^i = 1$ if the j-th instance is active and 0 if it is inactive. For example, for transistors $y_i = \sum_j q_j^i * W/L$, where W is the gate finger width and L is the length. For D components we can write $\mathbf{y} = [y_1, \cdots, y_D]$ and combine keys in a single key

$\mathbf{q} = [\mathbf{q}^1, \cdots, \mathbf{q}^D]$. Then, based on the m performances $\mathbf{p} = [p_1, \cdots, p_m]$ found in the datasheet, we can write m equations $p_j = \theta_j(\mathbf{y})$ linking each performance p_j to several y_i. An SMT-solver is used to find a key that satisfies all equations $p_j = \theta_j([\phi(\mathbf{q}^1), \cdots, \phi(\mathbf{q}^D)])$. The search space size is the same as in the brute-force attack, but with this approach we circumvent circuit simulations and we speed up the search. The difficulty with this approach is deriving the functions θ_j.

4) Hierarchical decomposition attack: For a complex system, to reduce the computational effort, the attacker may try to transform the extracted low-level netlist into a hierarchical, block-level representation and subsequently attack the circuit's sub-blocks individually. As mentioned in Section V, the simple defense is to obfuscate every single sub-block, but this is not strictly necessary. The reason is that sub-blocks are connected in feedback loops and only the global specifications are given in the datasheet, whereas many of the specifications of the sub-blocks are not released as they are not relevant for the end-user. We can imagine the scenario where obfuscation results in a circuit that has part of its sub-blocks obfuscated to a small degree such that the global specifications fail. We will see this obfuscation approach in the RF $\Sigma\Delta$ ADC case study in Section VIII. This scenario is confusing for the attacker as all sub-blocks are functioning correctly with an apparently decent performance trade-off, but the global performances are not met. Thus, the attacker cannot tell which sub-blocks have been obfuscated and all sub-blocks, even those that are left untouched by obfuscation, become candidates for de-obfuscation.

5) Automatic analog circuit sizing attack: We make the additional assumption that the attacker has access to a CAD tool for automatic analog circuit sizing. Such a tool starts with a given topology and aims at producing a sized topology that conforms to the performance objectives. An attacker may employ this tool to re-size the topology extracted from reverse engineering.

There exist several commercial CAD tools for automatic analog circuit sizing, for example the Optimizer in Eldo tool by Mentor Graphics, A Siemens Business, the WiCkeD tool by MunEDA, and the ID-Xplore by Intento Design. There are also several tools proposed in the literature (for example, see [31]–[37]).

To perform the sizing the attacker will need to define design variables and an objective function that measures the performance goal. To evaluate the objective function, the attacker will have to develop test benches for simulating the performances.

All these CAD tools require simulating the circuit at transistor-level a very large number of times. While this is possible for smaller circuit blocks, larger and complex circuits and systems, which are composed of several sub-blocks and have very long simulation times, cannot be handled as a single circuit. In this case, first a hierarchical decomposition of the circuit is needed. More specifically, the attacker will have to develop an abstract behavioral-level system model of the circuit that interconnects the sub-blocks and operates at data processing level, i.e. Simulink, VHDL-AMS, VerilogA, SystemC-AMS, etc. Having developed this system model, the attacker will need to guess sub-block performances to reach the global system-level performances since this information is lacking from the datasheet, as mentioned also in the hierarchical decomposition attack. With the guessed specifications, the attacker will launch the sizing tool to automatically size each sub-block at transistor level separately. Typically, the attacker will have to go through several iterations to meet the global performances using mixed-level simulations, where some sub-blocks are at transistor-level and some at behavioral-level.

Then, the next step is designing the layout. The attacker already has an extracted layout, but the automatically sized netlist will be different from the "deceivingly" sized reverse-engineered netlist. This is because many component sizing combinations achieve the same objective. Typically, the CAD tool will produce a Pareto front with several feasible solutions achieving different performance trade-offs. Therefore, the attacker will have to re-design large portions of the layout and change the floor-planing and routing. Typically, the attacker will have to do several design iterations going back and forth between schematic and layout, in order to meet post-layout performances.

In this regard, the attacker may rely on automated analog layout synthesis tools (for example, see [38]–[40]). However, these tools are not yet mature enough to produce first-time-right layout designs and require subsequent manual optimization to handle correctly symmetries, current flows, net parasitics, layout-dependent effects, etc. This is an active research area and there are no commercialized tools yet.

In short, most of the effort spent by an analog designer is not bypassed with this attack, with the exception that sub-blocks at transistor-level can be automatically sized at every design iteration. This attack requires a very high analog design expertise that goes far beyond the assumptions typically made on the capabilities of the attacker. In particular, the attacker will need to: (a) have knowledge on the use of automatic analog circuit sizing; (b) specify optimization objectives; (c) develop test benches for simulating performances; (d) develop an hierarchical behavioral-level model which is a challenging task on its own; (e) assign sub-block performances from target system-level performances; (f) have knowledge on analog layout design; (g) perform several design iterations that are driven by tough design decisions.

6) Physical attacks: These include: (a) optical imaging, i.e. using Scanning Electron Microscopy (SEM); (b) heat maps; (c) Focused Ion Beam (FIB)-assisted probing; and (d) electromagnetic (EM) side-channel analysis. As pointed out in [6], optical imaging would require first to narrow the search to the target obfuscated area so as to be able to extract such fine detail. However, the attacker has no means to pinpoint the obfuscated areas since every component is potentially an obfuscated one. Heat maps would not work either as they lack the necessary resolution to resolve the sub-gate-level inactive instances of an obfuscated component. With FIB-assisted probing the attacker will sequentially get access to all individual components to measure them and extract their sizing since every component is potentially an obfuscated one. This will be a very tedious and costly approach for large circuits, requiring several chips since FIB is destructive to the chip.

Regarding EM side-channel analysis, it is very unlikely to be able to resolve analog component sizings from the collected electromagnetic signals.

VII. Security metrics

Let us assume that the circuit has D components and that the i-th component has N_i instances out of which N_i^{obf} are inactive resulting from obfuscation. The *search space* for an attacker is defined as the number of all possible variants of the circuit:

$$S = \prod_{i=1}^{D} N_i. \tag{1}$$

However, the search space is in fact smaller for the following reasons: (a) certain components should be clearly matched and identical, for example the input transistor pair of an op-amp; (b) certain basic building blocks in the design are clearly replicated, i.e., switches, buffers, etc.; (c) the sizing of certain components may not be critical for setting the desired performance trade-off, i.e., this may be the case for digital control sub-blocks. Given these considerations, let O denote the set of components that are potentially obfuscated and let the cardinality of O be $|O| = D' \leq D$. This reduces the initial search space to:

$$S' = \prod_{i \in O} N_i. \tag{2}$$

This reduced search space S' is a metric of the hardness of reverse engineering. The attacker will try to reduce further the effective search space by making informed assumptions. In particular, the attacker knows that most likely the majority of components have not been obfuscated since otherwise this would have increased the obfuscation area overhead. In general, increasing the number of obfuscated components would make it more difficult to meet the intent design specifications. Specifically for the design flow in Fig. 1a, this would additionally require significant changes in the floor-planning and routing and, thereby, it would have been difficult to maintain a low performance penalty of the nominal obfuscated design with respect to the nominal non-obfuscated design. For this reason, the attacker would rather search using instance numbers close to the maximum value N_i. Let us assume that the attacker will try out the $\beta\%$ higher instance numbers for each potentially obfuscated component. This reduces the effective search space to:

$$S'' = \prod_{i \in O} \left\lceil \frac{\beta}{100} N_i \right\rceil. \tag{3}$$

The attacker can perform a brute-force analysis in this reduced search space in the hope of eventually guessing the correct sizing of the circuit.

Let us now define the parameters:

$$\alpha_i = \frac{N_i^{\text{obf}}}{N_i}, \tag{4}$$

$$\alpha_{\text{max}} = \max_i \alpha_i. \tag{5}$$

For the i-th component, the true number of active instances is $N_i - N_i^{\text{obf}}$, whereas the attacker will try out numbers of instances from $N_i - \left\lceil \frac{\beta}{100} N_i \right\rceil$ to N_i. Therefore, the attacker will "hit" the nominal sizing of the component during the search if $N_i - \left\lceil \frac{\beta}{100} N_i \right\rceil \leq N_i - N_i^{\text{obf}}$, which can be re-written as $\alpha_i \leq \left\lceil \frac{\beta}{100} \right\rceil$. Considering all components, the attacker will "hit" the nominal sizing of the circuit if $\alpha_{\text{max}} \leq \left\lceil \frac{\beta}{100} \right\rceil$. The parameter α_{max} is unknown to the attacker. The most favorable condition for the attacker is that he chooses exactly $\frac{\beta}{100} = \alpha_{\text{max}}$. Based on this most favorable condition, we define the following security metric λ_1 that *pessimistically for the defender* approximates the search space:

$$\lambda_1 = \log_2 \left(\prod_{i \in O} \lceil \alpha_{\text{max}} N_i \rceil \right). \tag{6}$$

The value of λ_1 is computed in bits to make it comparable to security levels from the digital domain.

We can define also a security metric λ_2 to express the total simulation time for an exhaustive search in the above reduced search space:

$$\lambda_2 = 2^{\lambda_1} \cdot T, \tag{7}$$

where T is the total simulation time for computing all performances using appropriate test benches.

We also acknowledge the possibility that circuit instances within the search space, other than the nominal circuit, may satisfy all specifications. For this reason, we define a security metric λ_3 to express their percentage:

$$\lambda_3 = 100 \cdot \frac{\sum_{j=1}^{n} I(j)}{n}, \tag{8}$$

where $n \leq 2^{\lambda_1}$ is the number of simulations that we afford to run and $I(j)$ is an indicator function with $I(j) = 1$ if the j-th circuit instance fails and $I(j) = 0$ otherwise.

Let now $\mathbf{p}_j = (p_{j1}, \cdots, p_{jk})$ denote the performance vector for the j-th circuit instance, where k is the number of performances, and let $\mathbf{s} = (s_1, \cdots, s_k)$ denote the specification vector. Other useful security metrics express in % the average deviation of failing circuits from specifications:

$$\lambda_4 = \frac{100}{n} \cdot \sum_{j=1}^{n} \|\mathbf{u} - \hat{\mathbf{p}}_j\|_2 \tag{9}$$

and the deviation of the "best" failing circuit that is closest to the specification boundary:

$$\lambda_5 = 100 \cdot \min_j \|\mathbf{u} - \hat{\mathbf{p}}_j\|_2, \tag{10}$$

where $\hat{\mathbf{p}}_j = (\hat{p}_{j1}, \cdots, \hat{p}_{jk})$, $\hat{p}_{ji} = \frac{p_{ji}}{s_i}$ if the j-th circuit fails the i-th specification and $\hat{p}_{ji} = 1$ if the j-th circuit passes the i-th specification, $\mathbf{u}$ is the $k \times 1$ vector with ones, and $\| \cdot \|_2$ is the L_2 norm. Note that $\|\mathbf{u} - \hat{\mathbf{p}}_j\|_2 = 0$ for passing circuits and $u_i - \hat{p}_{ji} = 0$ for passing performances.

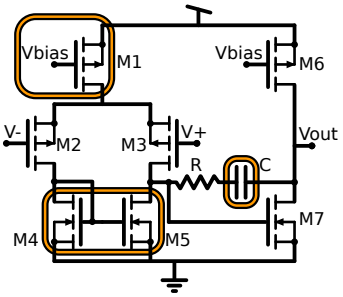


Fig. 6: Schematic of Miller op-amp. The obfuscated components are highlighted.

TABLE I: Design specifications and performance of nominal obfuscated and all-true contact designs.

Performance	Specs	Nominal obfuscated	All-true contact
Gain	$\geq$67 dB	70.1 dB	51.8 dB
GBW	$\geq$60 MHz	60.6 MHz	64.3 MHz
PM	$\geq$70°	71.2°	72.3°
THD	$\leq$0.1 %	0.04 %	5.1 %
I_{dc}	$\leq$400 µA	391 µA	416 µA

VIII. Case Studies

The proposed camouflaging methodology is demonstrated on two case studies, namely a Miller op-amp and an RF $\Sigma\Delta$ ADC. The Miller op-amp is a small basic building block and design guidelines can be found in textbooks. While not interesting for obfuscation as a stand-alone block, we provide this case study as a detailed and instructive example to illustrate also obfuscation metrics at block-level. The RF $\Sigma\Delta$ ADC is a large and complex circuit and demonstrates the true capabilities of the camouflaging methodology.

The simulation experiments were performed on an Intel(R) Xeon E5-2640 @ 2.5 GHz with 128 GB of RAM.

A. Miller Operational Amplifier

The Miller op-amp is designed in a 0.35μm CMOS technology following the design flow in Fig. 1b. Fig. 6 shows the schematic and the first two columns of Table I show the main performances and the target specifications.

We randomly obfuscated components to the point where we largely satisfied objective 1 while meeting objective 2. As shown in Table I, the performances of the nominal obfuscated design meet the target specifications, whereas the all-true contact design violates the Gain, Phase Margin (PM), power consumption (I_{dc}), and Total Harmonic Distortion (THD) specifications. In total, we iterated three times over the inner loop of Fig. 1b, and we did not have to repeat the inner loop during outer loop iterations for design optimization.

The obfuscated components include the biasing transistor M1, the current mirror transistors M4 and M5, and the feedback capacitor C, and are highlighted in the schematic in Fig. 6. M1 is laid out as a multi gate-finger transistor with 20 gate fingers out of which 10 are inactive. M4 and M5 are laid out in an interdigitized pattern and each has 12 gate fingers out of which 8 are inactive. Capacitor C is laid out as a capacitor bank with 4 unit capacitors out of which 1 is inactive. The obfuscated layout with this camouflaged sizing is illustrated in Fig. 7 highlighting the added inactive instances. The resultant obfuscation area overhead is 15%.

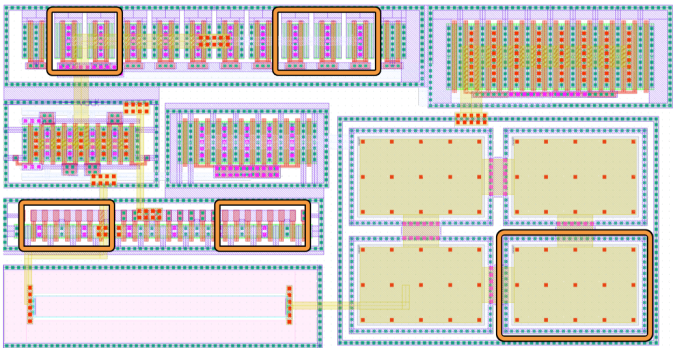


Fig. 7: Obfuscated layout of Miller op-amp highlighting the inactive instances that have been added.

TABLE II: Obfuscation of components in the Miller op-amp.

	M1	M2	M3	M4	M5	M6	M7	R	C
i	1	2	3	4	5	6	7	8	9
N_i	20	4	4	12	12	10	16	1	4
N_i^{obf}	10	0	0	8	8	0	0	0	1
α_i	$\frac{1}{2}$	0	0	$\frac{8}{12}$	$\frac{8}{12}$	0	0	0	$\frac{1}{4}$

Table II shows for each of the $D = 9$ components the total number of instances N_i, the number of obfuscated instances N_i^{obf}, and the parameter α_i. Out of these components, transistors M2 and M3 in the input differential pair are matched and transistors M4 and M5 in the current mirror are matched, thus $O = \{\text{M1}, \text{M2 or M3}, \text{M4 or M5}, \text{M6}, \text{M7}, \text{R}, \text{C}\}$ and $D' = 7$. The search space is computed from Eq. (2) to be $S' = 614400 \approx 2^{19.2}$. α_{max} is given by the current mirror transistors M4 and M5 and is computed to be $\alpha_{\text{max}} = 8/12 \approx 0.67$. Using these values Eq. (6) gives $\lambda_1 = 16.4$ bits. The simulation time to compute all performances is $T = 5$ seconds, thus $\lambda_2 = 120$ hours. Finally, we simulated a set of $n = 1000$ random variants of the circuit. None of them passed all the specifications, thus $\lambda_3 = 100\%$. The other two metrics evaluate to $\lambda_4 = 12600\%$ and $\lambda_5 = 2\%$. λ_4 turns out to be very high as for many circuit variants the THD is over 20% while it has an upper specification of 0.1%.

Table III summarizes the obfuscation metrics. In conclusion, involving camouflaging during the design flow did not increase design iterations and the nominal obfuscated design met the target specifications. Camouflaging with 15% area overhead resulted in a relatively high search space of 16.4 bits for such a small-size circuit, yet the brute force attack on an ideally reduced search space can be successfully completed in less than 120 hours.

B. RF $\Sigma\Delta$ ADC

We obfuscated an existing bandpass RF $\Sigma\Delta$ ADC design in a 65nm CMOS technology [41] following the design flow in Fig. 1a. Its block-level schematic is shown in Fig. 8. It is a large-size circuit with $D = 1100$ components and is

TABLE III: Obfuscation metrics for the Miller op-amp.

Security metrics								
S'	λ_1	λ_2	λ_3	λ_4	λ_5	Nominal obfuscated performance penalty	Obfuscation area overhead	All-true contact performance penalty
$2^{19.2}$	16.4 bits	120 h	100 %	12 600 %	2 %	no, see Table I	15 %	significant, see Table I

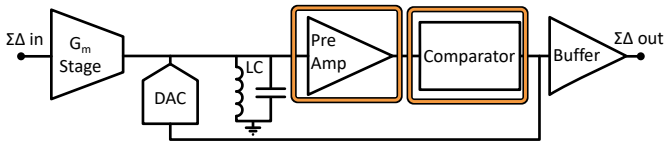


Fig. 8: Block-level diagram of the RF ΣΔ ADC. Obfuscated sub-blocks are highlighted.

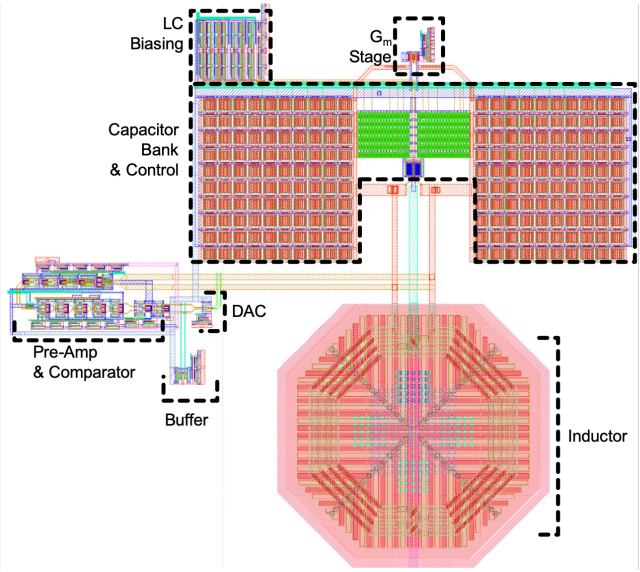


Fig. 9: Complete layout of the RF ΣΔ ADC.

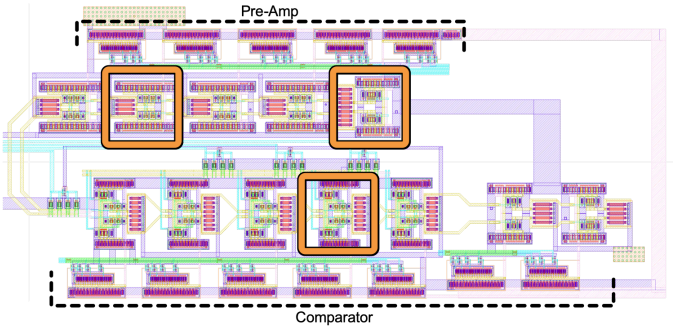


Fig. 10: Zoom in into the pre-amplifier and comparator layouts. The obfuscated blocks are highlighted.

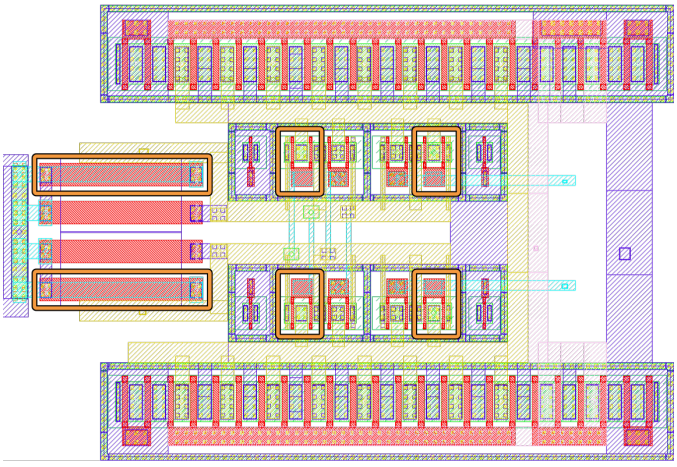


Fig. 11: Zoom in into the obfuscated areas of one amplification stage of the pre-amplifer. The obfuscated areas are highlighted.

composed of several sub-blocks. It is part of an RF receiver and is re-configurable such that the RF receiver can be programmed to serve for establishing communication using several standards within the frequency range from 1.5 GHz to 3 GHz, including Bluetooth, ZigBee, WiFi 802.11b, LTE1800, LTE2100, LTE2600, etc. Herein, we consider a fixed configuration setting where the center frequency of the bandpass ΣΔ ADC is set at f_0 =3 GHz and the sampling frequency is set at f_s =12 GHz.

A careful look at the circuit netlist shows that $D' = 75$ components are candidates for obfuscation. To satisfy objective 1 we iterated three times over the inner loop of Fig. 1a, and then to satisfy objective 2 we had to iterate once over the outer loop of Fig. 1a. The two objectives were met by obfuscating a few components in only two of the sub-blocks, namely the pre-amplifier and the comparator, as illustrated in Fig. 8. In particular, within the pre-amplifier we obfuscated two differential transistor pairs and a resistor in two different amplification stages, and within the comparator we obfuscated a latch through its input differential transistor pair. The differential transistor pairs are laid out in common-centroid pattern and the resistor in a serpentine pattern. Fig. 9 shows the obfuscated layout. Fig. 10 zooms in into the obfuscated pre-amplifier and comparator. Fig. 11 shows a further zoom in into the obfuscated areas of one amplification stage of the pre-amplifer. The obfuscation area overhead is practically zero.

We consider the main performance which is the Signal-to-Noise Ratio (SNR). Fig. 12 shows the SNR as a function of input power amplitude for the nominal non-obfuscated, nominal obfuscated, and all-true contact designs. The SNR is computed on the layout extracted netlist with parasitics. One approximate SNR simulation for a given input power amplitude took up roughly 5 hours. As it can be seen, the nominal obfuscated design shows no performance penalty, whereas the all-true contact design shows a degraded SNR that even falls below 0 dB for smaller power amplitudes, which means that the signal is completely buried under noise. In fact, the small obfuscation within the pre-amplifier and

TABLE IV: Obfuscation metrics for the RF ΣΔ ADC.

Security metrics						Nominal obfuscated performance penalty	Obfuscation area overhead	All-true contact performance penalty
S'	λ_1	λ_2	λ_3	λ_4	λ_5			
2^{139}	110 bits	2.5×10^{29} years	100 %	139.11 %	8.56 %	no, see Fig. 12	0 %	significant, see Fig. 12

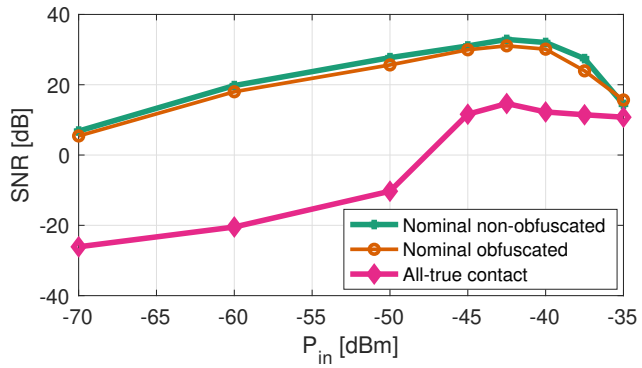


Fig. 12: SNR vs. input power amplitude for the nominal non-obfuscated, nominal obfuscated, and all-true contact designs.

comparator resulted in slight performance deviation for these two sub-blocks. Since the specifications of the sub-blocks are unknown to the attacker, the attacker cannot identify which are the obfuscated sub-blocks.

In total, the search space is computed from Eq. (2) to be $S' = 8.9 \times 10^{41} \approx 2^{139}$. α_{max} is given by an obfuscated differential transistor pair in a pre-amplifier stage and is computed to be $\alpha_{\text{max}} = 8/12 \approx 0.67$. Using these values Eq. (6) gives $\lambda_1 = 110$ bits. Since simulating the extracted layout netlist is very time-consuming, we will assume that the attacker will perform a first step analysis at schematic-level where one approximate SNR simulation for a given input power amplitude takes up far less time, about 20 minutes. We consider that the attacker will measure SNR at 5 input power amplitudes $P_{in} = \{-60, -50, -40, -37.5, -35\}$ dBm spanning the input dynamic range, thus total simulation time will be 100 minutes. In fact, the attacker will have to verify additional performances, e.g. Spurious Free Dynamic Range (SFDR). Assuming $T \geq 100$ minutes as an optimistic lower bound of simulation time per circuit instance, it still gives $\lambda_2 \geq 2.5 \times 10^{29}$ years. Due to the costly simulations, we simulated $n = 100$ random variants of the circuit and none of them passed the SNR specification, giving $\lambda_3 = 100\%$. We computed also $\lambda_4 = 139.11\%$ and $\lambda_5 = 8.56\%$.

Table IV summarizes the obfuscation metrics. In conclusion, the practically zero-overhead obfuscation resulted in no measurable performance penalty for the nominal obfuscated design, in significant performance penalty for the all-true contact design, and in utterly impossible reverse engineering via a brute force attack on an ideally reduced search space.

IX. Conclusion

We presented an obfuscation methodology for analog ICs via sizing camouflaging making use of fake contacts. We proposed two camouflaging design flows that consider camouflaging of an existing design and involving camouflaging in the design phase. We demonstrated that for realistic and large-size circuits and systems the methodology results in remarkable security against a brute-force attack performed in a reduced space after some informed assumptions by the attacker. We demonstrated also that it suffices to obfuscate few components, which minimizes the overall camouflaging effort and yields practically zero area overhead and performance penalty. In terms of future work, we are planning to extend the library of obfuscated components and also study more extensively possible counter-attacks sketched in Section VI.

Acknowledgments

This work has been carried out in the framework of the ANR STEALTH project with N° ANR-17-CE24-0022-01. It is partially funded by the ANR TOLTECA project with N° ANR-16-CE04-0013-01. J. Leonhard has a fellowship from the doctoral school EDITE de Paris.

References

[1] 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.

[2] 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.

[3] 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.

[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] B. Colombier and L. Bossuet, "Survey of hardware protection of design data for integrated circuits and intellectual properties," *IET Computers & Digital Techniques*, vol. 8, no. 6, pp. 274–287, 2014.

[6] 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.

[7] 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.

[8] M. Yasin, J. Rajendran, and O. Sinanoglu, *Trustworthy Hardware Design: Combinational Logic Locking Techniques*, Springer, 2020.

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

[10] 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.

[11] 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.

[12] R. P. Cocchi, J. P. Baukus, L. W. Chow, and B. J. Wang, "Circuit camouflage integration for hardware IP protection," in *Proc. IEEE/ACM Design Automation Conference*, 2014.

[13] S. Chen, J. Chen, D. Forte, J. Di, M. Tehranipoor, and L. Wang, "Chip-level anti-reverse engineering using transformable interconnects," in *Proc. IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems*, 2015, pp. 109–114.

[14] 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.

[15] S. Patnaik, M. Ashraf, J. Knechtel, and O. Sinanoglu, "Obfuscating the interconnects: Low-cost and resilient full-chip layout camouflaging," in *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 2017, p. 41–48.

[16] 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.

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

[18] 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.

[19] 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.

[20] N. G. Jayasankaran, A. Sanabria Borbon, A. Abuellil, E. Sánchez-Sinencio, J. Hu, and J. Rajendran, "Breaking analog locking techniques via satisfiability modulo theories," in *Proc. IEEE International Test Conference*, 2019, Paper 9.1.

[21] N. G. Jayasankaran, A. S. Borbon, E. Sanchez-Sinencio, J. Hu, and J. Rajendran, "Towards provably-secure analog and mixed-signal locking against overproduction," in *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 2018.

[22] S. Govinda 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 and Test in Europe Conference*, 2020.

[23] 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 and Test in Europe Conference*, 2020.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] Y. Tsividis, *Mixed Analog-Digital VLSI Devices and Technology*, World Scientific, 2002.

[29] M. El Massad, S. Garg, and M. Tripunitara, "Integrated circuit (IC) decamouflaging: Reverse engineering camouflaged ICs within minutes," in *Proc. Network and Distributed System Security Symposium*, 2015.

[30] C. Yu, X. Zhang, D. Liu, M. Ciesielski, and D. Holcomb, "Incremental SAT-based reverse engineering of camouflaged logic circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 10, pp. 1647–1659, 2017.

[31] D. M. Binkley, C. E. Hopper, S. D. Tucker, B. C. Moss, J. M. Rochelle, and D. P. Foty, "A CAD methodology for optimizing transistor current and sizing in analog CMOS design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 2, pp. 225–237, 2003.

[32] W. Daems, G. Gielen, and W. Sansen, "Simulation-based generation of posynomial performance models for the sizing of analog integrated circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 5, pp. 517–534, 2003.

[33] B. Liu, Y. Wang, Z. Yu, L. Liu, M. Li, Z. Wang, J. Lu, and F. V. Fernández, "Analog circuit optimization system based on hybrid evolutionary algorithms," *Integration*, vol. 42, no. 2, pp. 137 – 148, 2009.

[34] T. McConaghy, P. Palmers, P. Gao, M. Steyaert, and G. Gielen, *Variation-Aware Analog Structural Synthesis*, Springer, 2009.

[35] T. Y. Zhou, H. Liu, D. Zhou, and T. Tarim, "A fast analog circuit analysis algorithm for design modification and verification," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 2, pp. 308–313, 2011.

[36] A. Malak, Y. Li, R. Iskander, F. Durbin, F. Javid, J.-M. Guebhard, M.-M. Louërat, and A. Tissot, "Fast multidimensional optimization of analog circuits initiated by monodimensional global peano explorations," *Integr. VLSI J.*, vol. 48, no. C, pp. 198–212, 2015.

[37] Y. Li, Y. Wang, Y. Li, R. Zhou, and Z. Lin, "An artificial neural network assisted optimization system for analog design space exploration," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2019.

[38] H. E. Graeb (Ed.), *Analog Layout Synthesis: A Survey of Topological Approaches*, Springer, 2011.

[39] N. Lourenço, R. Martins, A. Canelas, R. Póvoa, and N. Horta, "AIDA: Layout-aware analog circuit-level sizing with in-loop layout generation," *Integration*, vol. 55, pp. 316 – 329, 2016.

[40] H. Ou, K. Tseng, J. Liu, I. Wu, and Y. Chang, "Layout-dependent effects-aware analytical analog placement," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 8, pp. 1243–1254, 2016.

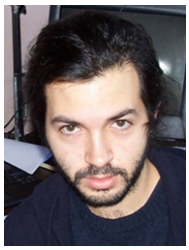
[41] A. Sayed, T. Badran, M.-M. Louërat, and H. Aboushady, "1.5-to-3.0 GHz tunable RF $\Sigma\Delta$ ADC with a fixed set of coefficients and a programmable loop delay," *IEEE Transactions on Circuits and Systems - II: Express Briefs*, 2020, to be published.

Julian Leonhard (S'17) received the B.Sc. and M.Sc. in electrical and computer engineering from the Technical University of Munich, Germany, in 2016. He was a working student and wrote his Bachelor thesis with Intel Mobile Communication, Munich. For his Master Thesis he worked together with Infineon Technologies, Munich. Since 2017 he is a PhD candidate at LIP6 Laboratory, Sorbonne Université in Paris, France. His research topics include hardware security and design for trust for analog and mixed-signal circuits.

Alhassan Sayed received the B.Sc. and the M.Sc. degrees in Electrical Engineering from the Electronics and Communications Department of Minia University, Minia, Egypt, in 2007 and 2010, respectively. He obtained his Ph.D. degree in Electrical Engineering and Computer Science from Sorbonne University, Campus Pierre and Marie Curie, 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, Egypt. He is also with Seamless Waves Semiconductor, a spin-off company from Sorbonne University, Paris, France. 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 (S'97-M'02) received the B.Sc. degree in Electrical Engineering from Cairo University, Egypt, in 1993, and the M.Sc. and Ph.D. degrees in Electrical Engineering and Computer Science from Sorbonne University, Campus Pierre & Marie Curie, Paris, France, in 1996 and 2002, respectively. He also obtained his accreditation to supervise research (HDR) from the same University in 2010. He is currently an Associate Professor at Sorbonne University, Campus Pierre & Marie Curie, Paris, France. He worked on the design of high resolution audio Digital-to-Analog converters at Philips Research Laboratories (currently NXP), Eindhoven, The Netherlands. He also worked on the implementation of a baseband continuous-time Sigma-Delta modulator for RF receivers at STMicroelectronics, Crolles, France. He was a visiting professor for several months at the French University in Egypt, UFE, the Federal University of Rio Grande do Norte, UFRN, Brazil, and the "Technologico de Monterrey", ITESM, Guadalajara, Mexico, in 2007, 2011 and 2013 respectively. During the academic year 2012-2013, he was on a sabbatical leave at the Ecole Polytechnique, LIPCM laboratory, working on the design of analog circuits using organic electronics. 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 (S'02-M'07) received the Diploma in electrical and computer engineering from the National Technical University of Athens, Greece, in 2001 and the Ph.D. in electrical engineering from Yale University, USA, in 2006. From October 2007 to May 2015 he was a Researcher with the French National Center for Scientific Research (CNRS) at TIMA Laboratory, Université Grenoble Alpes, Grenoble, France. Currently he is a Researcher with the 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).