Journal of Electronic Testing (2021) 37:653–673
https://doi.org/10.1007/s10836-021-05976-8

Reducing Aging Impacts in Digital Sensors via Run-Time Calibration

Md Toufiq Hasan Anik[1] · **Mohammad Ebrahimabadi**[1] · **Jean-Luc Danger**[2] · **Sylvain Guilley**[2] · **Naghmeh Karimi**[1]

Received: 8 May 2021 / Accepted: 11 November 2021 / Published online: 1 February 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
Hazards or intentional perturbations must be identified in safety- and security-critical applications. Digital sensors have been shown to be an appealing approach to detect such abnormalities. However, as any sensor technology, digital sensors are prone to mis-calibration. In particular, even if the digital sensor initial calibration is correct, the rate of false and missed alarms might increase when the sensor is aged. In this paper, we thoroughly study the impact of aging-induced false and missed alarms. Indeed aging relates to the usage time, and *a priori* model (historical data for environmental variation) for predicting the aging is unrealistic for digital sensors as tracking the usage time with related temperature and voltage variation imposes high overhead. Accordingly, we propose an alternative approach where not one but two sensors are deployed. In practice, one sensor is used to detect environmental deviations, while the second one is used as the reference. In this respect, the second sensor is only operated seldom, mostly to re-calibrate the active sensor when aged. From this dual input (unaged and aged sensor), corrective models are derived. We account for two methods, namely simple but effective offset correction, and adjustment based on machine-learning. We conduct extensive characterizations (both pre-silicon simulations and post-silicon measurements on FPGA) which quantitatively confirm the applicability and high sensitivity of digital sensors.

Keywords Digital sensors · Calibration · Missed & false alarms · Device aging · Anomaly detection · Machine learning · Device recalibration

1 Introduction

Ensuring reliability and security of integrated circuits is a growing challenge in industry due to the rising competition in the production market as well as the increasing demand for high-quality electronic goods. Optimized-performance and reduced-power demands resulted in shrinking the feature size and in turn enabled a single chip to include billions of transistors. Ensuring the reliability and security of such complex circuitries is of utmost importance [1, 2].

Practically, during the chip design phase, well-defined environmental conditions in which a chip can operate without experiencing any failure and/or compromising its security is drawn [3, 4]. These conditions relate to the proper range of operating temperature, voltage, and manufacturing process so-called PVT altogether. Accordingly, the foundries set out the PVT corners [5] under which the circuit can operate nominally.

Although chips are designed to work under nominal conditions, they may be exposed to different pressures, such as very high/low temperatures or over/under power supply where the intended PVT that has been defined for the chip at the design time is violated. Such violation can be unintentional or intentional. For instance, unintentional violations can occur in the circuitries residing close to the explosion or electric engine in automotive products. Another example is that of mission-critical chips operating in space or nuclear power plants where they are exposed to high magnetic fields or various irradiations.

Responsible Editor: O. Sinanoglu

Md Toufiq Hasan Anik and Mohammad Ebrahimabadi contributed equally to this work.

✉ Md Toufiq Hasan Anik
toufiqhanik@umbc.edu

Mohammad Ebrahimabadi
ebrahimabadi@umbc.edu

Jean-Luc Danger
jean-luc.danger@telecom-paristech.fr

Sylvain Guilley
sylvain.guilley@secure-ic.com

Naghmeh Karimi
nkarimi@umbc.edu

[1] CSEE Department, University of Maryland Baltimore County, Baltimore, MD 21250, USA

[2] Think Ahead Business Line of Secure-IC S.A.S. & Institut Polytechnique de Paris, Paris, France

In contrast, an intentional violation can be due to a malicious attack [6, 7] aiming at a denial of service, malfunctioning, or even leaking sensitive data [8]. Attacks can be either realized with a physical access to the targeted chip or remotely if the chip has a remote login facility [9–11]. For example, remotely hijacking the chip operation, running a malicious code forcing the core to increase performance for raising its temperature beyond the Temperature (T) corner, and operating the DVFS management system in turbo mode (that is: "over-clocked") possibly with decreased Voltage supply (V) all can violate VT corners. CLKSCREW [12–14], PlunderVolt [11], VoltJockey [15], V0LTpwn [16], FPGA-hammer [10], RAM-Jam [9], and VoltPillager [17] are some representative examples of such attacks.

An attack's impact can be highly catastrophic for the mission-critical chips [18], e.g., for the chips used in medical applications as they may lead to a life loss situation. As another example one can point to the malicious PVT violations on cryptographic devices that may result in sneaky information leakage [19, 20]. Indeed, the cryptographic chips are leveraged to preserve the security of sensitive data. However, they can leak the secret information, e.g, keys, by fault analysis [21] related to the PVT violations [8]. Consequently, it is necessary to sense the operating conditions such as voltage and temperature for optimized performance as well as tracking (preventing in some cases) the anomalies, malfunction, and malicious attacks. In practice, embedded sensors are used to notify the users by raising alarms while chips are operating out of specification violating the PVT corners [2], thus enhancing the chips' reliability and security.

Conventionally, analog sensors have been broadly used to sense the operating conditions and to raise alarms when defined VT is not met. However, such sensors suffer from different shortcomings such as costly post-fabrication calibration due to process variation, difficulty for adaptation to new technology nodes, low portability to thinner technologies, and most importantly high false alarm rate. As an example, brownout sensors only detect lower than expected power supply, but might not react to transient power glitches or combination of power and clock glitches. Therefore, a suitable replacement for analog sensors is needed. On the other hand, due to their low-cost design, effortless and easy adaptation to the advanced technology nodes, high portability, and optimized performance digital sensors have received the lion's share of attention in the recent years and have been shown as a promising solution [1]. Indeed digital sensors were presented in low-power literature (e.g., for fine-tuning the Dynamic-Voltage-Frequency-Scaling [22]), and in the security-related literature [24, Fig. 14, page 189] in 2011. Thereafter, they were adopted by the industry [23] and government sectors.

Instead of measuring physical quantities (such as temperature and voltage) separately, digital sensors are intended to detect functional unintentional and/or malicious failures [25] by sensing different operating quantities such as Process (P), Voltage (V), and Temperature (T) altogether without precise knowledge about each. This results in fewer false alarms raised by digital sensors compared to their analog counterparts [1]. Although digital sensors seem a promising solution to detect anomalies when the device is new, they suffer from taking the device aging effects into account [3]. Such phenomena that occur during the circuits' operation over time results in missed or false alarms, i.e., inability to detect an existing anomaly or raising alarms in the absence of anomalies, respectively.

In this paper, we consider the device aging effects and propose two low-cost calibration schemes that can be used during the circuits' run-time to decrease the number of false/missed alarms. We named the first one as Differential Calibration (DC) and next one as Machine Learning Based Calibration (ML-DC).

The first scheme, Differential Calibration (DC) is performed with a low-cost differential method benefiting from a digital sensor and its replicated counterpart where the replicated sensor turns on rarely, thus is affected much less by aging. Using the differential output across the main sensor and the replicated one, the outcome of the main sensor is calibrated during the runtime to take the aging impacts into account

To enhance the performance of the proposed Differential Calibration (DC) scheme and to reduce its missed alarm rates even more, we deploy Machine Learning Based Calibration (ML-DC), which uses Machine Learning (ML) techniques, and in particular Linear Regression (LR) scheme, due to its low overhead, in our second approach. Here, we use the DC scheme on top of the targeted ML scheme to find a more appropriate value to be used for calibration of the main sensor during runtime. In this method, the last N readings (N is as low as 4 in our experiments) of the main and replicated sensors are used during the calibration process. The contributions of this research are as follows.

- A run-time differential calibration scheme to reduce aging-induced missed/false alarms in the targeted digital sensor over time;
- A hybrid approach that combines our differential calibration scheme with machine learning methods to enhance the accuracy of the sensor in detecting anomalies and attacks during the runtime;
- Thorough investigation of the tradeoffs on characterizing sensors;
- Hardware implementation of the sensor characterisation and the alarm-rising circuitry;

– Detailed HSpice simulations to investigate the effect of aging on the original sensor versus the sensors equipped with the proposed run-time calibration schemes;
– Investigation of the outcome of the targeted sensor in the FPGA fabrics.

Outline. The rest of the paper is organized as follows. Section 2 discusses the threat model and the motivation of this research. Section 3 presents the preliminary backgrounds on digital sensors and aging mechanisms. Section 4 gives a description of the digital sensor targeted in this study and its characterization. Section 5 presents our proposed calibration techniques. Section 6 details the hardware implementation of the sensor characterization and the alarm-raising circuitry. Experimental setup and results are discussed in Sect. 7. Our FPGA implementation and the related outcome are discussed in Sect. 8. Finally, the conclusion and future directions are summarized in Sect. 9.

2 Threat Model and Motivation

2.1 Threat Model

From a historical perspective, secure chips used to be of "smartcard" form factor, where sensitive signals (clock, reset, power) were provided externally, via dedicated pins (recall ISO/IEC 7816-2). Thus attackers could easily manipulate those signals to disrupt the secure chip normal operation, respectively through overclocking, partial DFF reset, and underfeeding.

Modern systems are embedded as systems-in-package (SiP), and the current trend is even for a tighter integration in a systems-on-chip (SoC). In SoC architectures, the clock is filtered through internal Phased-Lock Loops (PLLs). This means that attempts to introduce surreptitious glitches on the external clock happen to be filtered out by the PLL, acting as a low-pass filter. The same situation happens with the reset line, which is controlled by a Power-On-Reset (POR) system that aims at preventing glitchy resets. As far as power supply is concerned, local decoupling capacitances (spread on the PCB and even on the chip) contribute to maintain an operational voltage despite some brownout show up. For all these reasons, attack vector arising from fast varying inputs are less and less likely to happen in SoCs.

Indeed, all attempts to change in a slow manner the chip "operational environment" (power, temperature, clock, etc.) are likely to be reflected within the SoC. In particular, the recent "DVFS abuse" attacks (listed in the introduction Sect. 1) are crucial. Their common point is that they move the chip from a safe to a dangerous operational state inconspicuously. For instance, regarding the power, it is well-known that with nowadays very low core voltages (usually $< 1\ V$), any IR drop is seen as an aggression. Therefore, the threats we consider in this paper are those which consists in slightly drifting apart from nominal PVT conditions. In addition, we also consider more brutal threats, which (all of the sudden – albeit in the "slow motion" pace discussed above) alter the power/temperature/time reference. For instance, PlunderVolt does lead (step by step) to unexpected system-level IR drops. In general, all recent "DVFS attacks" consist in placing the chip slightly out of bounds, and happen to be successful.

As a motivating example, NIST FIPS 140-3 [26, Clause 7.7.4.3] mandates tests within (*stable*) temperature range of $[-100°C, +200°C]$, which is by far a larger interval compared to even the most stringent corners in consumer applications.

Notice that fast and local attacks, can be produced by focused lasers shots. Nonetheless, we notice that SoCs are themselves inserted in SiP. Typically, applicative SoCs are sandwiched between FLASH memory (on the one side) and RAM memory (on the other side). Shooting a focused laser through FLASH/RAM in a reproducible manner is considered bespoke as of today, hence such accurate and fast attacks are simply not considered in our threat model.

In sum, our threat model considers the cases in which the voltage (power), temperature, or the internal clock is changed.

2.2 Motivation: Impact of Aging on Digital Sensors

In practice, sensors are deployed to detect the deviation from the nominal operating conditions. But when a digital circuit is used for a while (i.e., aged), its electrical specifications change. Accordingly as will be discussed in Sect. 3.3, its underlying components become slower. Similarly, the digital sensors we target in this paper are affected by device aging. This results in inaccuracies in firing alarms, i.e., an aged sensor may raise an alarm in cases that are not supposed to (the so-called false alarms) or may not raise an alarm when it should (the so-called missed alarm). Both missed alarms and false alarms must be carefully taken into account. The former influences security and the latter triggers availability issues. Accordingly this paper targets the discussed digital sensors and opts to improve their accuracy in raising alarms by decreasing the rate of false and missed alarms during the run-time.

3 Preliminary Backgrounds

3.1 Background on Digital Sensors

Delay-based digital sensors are realized via artificially inserting a critical path in the target system. Such path is monitored regularly (ideally at every clock cycle) regarding setup time violation. The absence or presence of such

violation reveals information about the current environmental condition as the gate delays are impacted by these conditions. Such sensing applies clearly on the sensor logic itself, but by extension also to its surrounding logic. This is at the gist of digital sensors: what they sense represents a comparable status compared to the neighboring gates (which make up the part protected by the digital sensor). Such assumption is safe in practice, as it is for instance the basic hypothesis in "bundled-data" asynchronous circuits [27]. Indeed, in such circuits, the control and the data paths are also expected to track in terms of timing.

3.2 Digital versus Analog Sensors

Analog sensors have been used for a long time as transducers that quantify the ambient conditions of electronic devices. These sensors, however, suffer from various flaws that have received considerable attention in recent years [1, 2]. On the other hand, digital sensors have appeared as highly qualified counterparts concurring such drawbacks. What follows discusses the benefits of digital sensors compared to the analog alternatives.

Analog sensors require complete custom layout design and an expensive calibration phase after manufacturing [28]. Digital sensors, on the other hand, are composed entirely of digital standard cells and do not require costly calibration [24, Fig. 14, page 189]. Their customized structure makes the analog sensors less portable while digital counterparts, thanks to their underlying standard cells, are conveniently portable. As the Physical Design Kit (PDK) hardware is revised, analog sensors need revalidation by new simulations. In contrast, digital sensors merely require simple recalibration in each of these cases by adjusting the length of the delay chain.

Digital sensors are mainly optimized benefiting from the optimization process during the RTL and gate-level synthesis via commercial EDA tools whereas analog alternatives usually experience manual dimensioning. Analog sensors rely on "always-on" logic gates while digital sensors are more controllable, benefit from clock gating, and only consume power during toggling [29]. Digital sensors can be calibrated to work in different Dynamic Voltage/Frequency Scaling (DVFS) configurations.

The digital sensors are sensitive to supply voltage and temperature altogether without precious knowledge of each. This makes digital sensors much smarter; producing fewer false alarms compared to the analog counterparts that consider the physical quantity of supply voltage and temperature separately and thus fail to consider that the effect of high temperature may be compensated in high voltages [1]. Both digital and analog sensors suffer from process variation and dynamic noise. While analog sensors counter ambiguities in defining a threshold for nominal vs abnormal situations, the digital sensors resolve this issue via electrical level discretization [30].

In practice, as digital sensors detect environmental changes fast, they are suitable for both slow-stress (e.g., global perturbation [31]) and transient attacks (e.g., glitches, or local electromagnetic field/laser light injections).

3.3 Background on Aging

To measure abnormal environmental conditions different digital sensors have been proposed in recent years, e.g. [24, Fig. 14, p. 189], [33, Fig. 3, p. 441], [32] all of which share the property of being realized only by simple digital gates including D Flip-Flops (DFFs), buffers, etc, and of having maximal activity, thereby (as a drawback) being especially prone to aging.

Aging mechanisms result in performance degradation and eventual failure of digital circuits over time. Among aging mechanisms, Negative Bias Temperature-Instability (**NBTI**) and Hot-Carrier Injection (**HCI**) have been shown to be more prominent in CMOS technologies [34], resulting in increasing switching and path delays.

NBTI Aging: NBTI affects PMOS transistors. Indeed, a PMOS transistor experiences two phases of NBTI depending on its operating condition. The first phase, the so-called stress phase, occurs when the transistor is on ($V_{gs} < V_t$). Here, positive interface traps are generated at the Si-SiO_2 interface which lead to an increase of the threshold voltage of the transistor. The second phase, the so-called recovery phase, occurs when the transistor is off ($V_{gs} > V_t$). The threshold voltage drift that occurred during the stress phase will partially recover in the recovery phase. Threshold voltage drifts of a PMOS transistor under stress depend on the physical parameters of the transistor, supply voltage, temperature, and stress time [36, 37]. The last three parameters (so-called external parameters) are used as acceleration factors of aging process. Figure 1 shows the threshold voltage drift of a PMOS transistor that is continuously under stress for 6 months and a transistor that alternates stress/recovery phases every other month. As shown, the NBTI effect is high in the first couple of months but the threshold voltage tends to saturate for long stress times.

HCI Aging: HCI mainly occurs in NMOS transistors when hot carriers are injected into the gate dielectric during transistor switching and remain there. HCI is a function of switching activity and degrades the circuit by shifting the threshold voltage and the drain current of transistors under stress. HCI-induced threshold voltage drift is sensitive to the number of transitions occurring in the gate input of the transistor. In fact, HCI has a dependency on temperature, clock frequency, usage time, and activity factor of the transistor under stress, i.e., the percentage of cycles in which the transistor is switching [34].

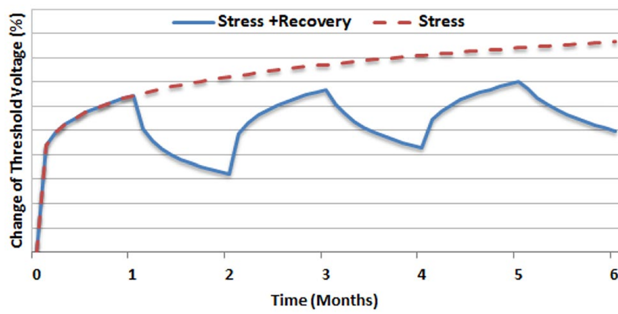


Fig. 1 Threshold-voltage shift of a PMOS transistor under NBTI effect [35]. Values on Y axis are not shown to make the graph generic across different silicon foundries and technological nodes

4 Target Sensor and its Characterization

4.1 Digital Sensor Preliminaries

The target digital sensor is realized via inserting an artificial critical path, referred to as "Delay Chain" hereafter. The timing conditions of such a path are affected by the operating conditions thus can be leveraged to ensure if the underlying circuitry is operating under the PVT specifications it is supposed to. This delay chain (shown in Fig. 2) is fed by a rising or a falling transition and is verified if such an edge manages to propagate at the considered clock frequency to the end of the chain ([24, Fig. 14]). The inability to do so is indicative of environmental disruption or exploitation. During an irregular process, the setup time violation happens in the very first place on the artificial critical path and is monitored by the digital sensor by raising an alarm to call for proper action to be taken. To be able to sample the delay chain and characterize the amplitude of the timing violation and thereby digitize the amount of stress added to the circuit, a number of flip-flops are introduced into various parts of this delay chain.

Figure 2 illustrates the digital sensor architecture implemented in this study [38]. It includes $n_0 + n_1$ buffers where the last n_1 buffers are connected to individual D flip-flops for sampling the timing from the delay chain. All flip-flops are operating at the frequency of F. A periodic signal with a frequency of $F/2$ is generated by a Toggle flip-flop (T Flip-Flop) feeding the first buffer, and is propagated through the

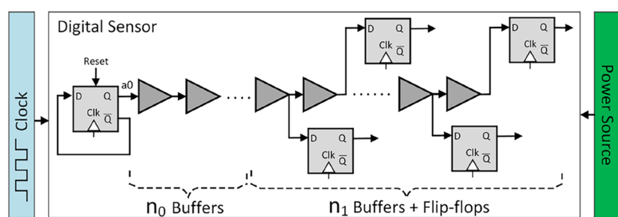


Fig. 2 Targeted digital sensor architecture

delay chain. The outputs of the included flip-flops are collected (internally) to characterize the sensor outcome. When timing requirement is not met the sensor raises an alarm. Note that the timing requirement and sensor's design (number of buf fers and flip-flops) varies with the operational range of the sensor. Indeed, chips are commonly manufactured in multiple temperature classes (e.g. consumer, automotive, military, etc.) depending on their uses, each considering a particular temperature range under which the chip is supposed to be usable. The digital sensor design methodology and the process to architect the sensor, i.e., deciding about the number of flip-flops and buffers during the sensor design procedure, is discussed in details in [1].

4.2 Preliminaries on Characterization

The advantage of the digital sensor is that its output depends on the voltage and temperature quantities as a whole, and not their individual quantities. This is important since for example increasing the temperature beyond the expected range may be compensated with a higher voltage [4]. Indeed, deviation of voltage and temperature from their nominal range altogether can affect the timing requirement of the delay chain embedded in the digital sensor, and in turn can manifest intentional or unintentional operating failures.

To characterize the digital sensor shown in Fig. 2, a parameter so-called Average Flip-Flop Number (AFN) is used [3]. In each voltage and temperature combination, noted as (V, T) hereafter, the AFN is derived based on the flip-flops' outputs. Indeed, depending on the voltage and temperature quantities, the delay chain's propagation delay varies, thereby different sets of values are captured by the flip-flops in different (V, T) configurations.

As this sensor is supplied with the $a0$ periodic signal, in each clock cycle (CC_i) the first $FN_i - 1$ flip-flops would be in phase A (say for example $0 \rightarrow 1 \rightarrow 0$) and the remaining flip-flops would be in the complementary phase $\overline{\text{A}}$ (say $1 \rightarrow 0 \rightarrow 1$), where FN_i corresponds to the flip-flop index at which step $\overline{\text{A}}$ begins in the clock cycle CC_i. Note that this FN_i index changes in different (V, T) configurations. For characterization, the average of all FN_i values (each related to one clock cycle) is evaluated. This average is called AFN. It is noteworthy to mention that the AFN value would be lower when the circuitry (as well as the delay chain) operates slower (under a high temperature and/or low voltage). The AFN tends to higher indexes when the circuit operates faster (under a low temperature and/or high voltage). An alarm is raised when the circuit operates out of the specifications it was designed for, i.e., when the circuit operates slower or faster than expected. Such conditions result in an AFN value that is out of bound (lower than the minimum AFN or higher than the maximum AFN related to the circuit operating conditions based on its specifications.)

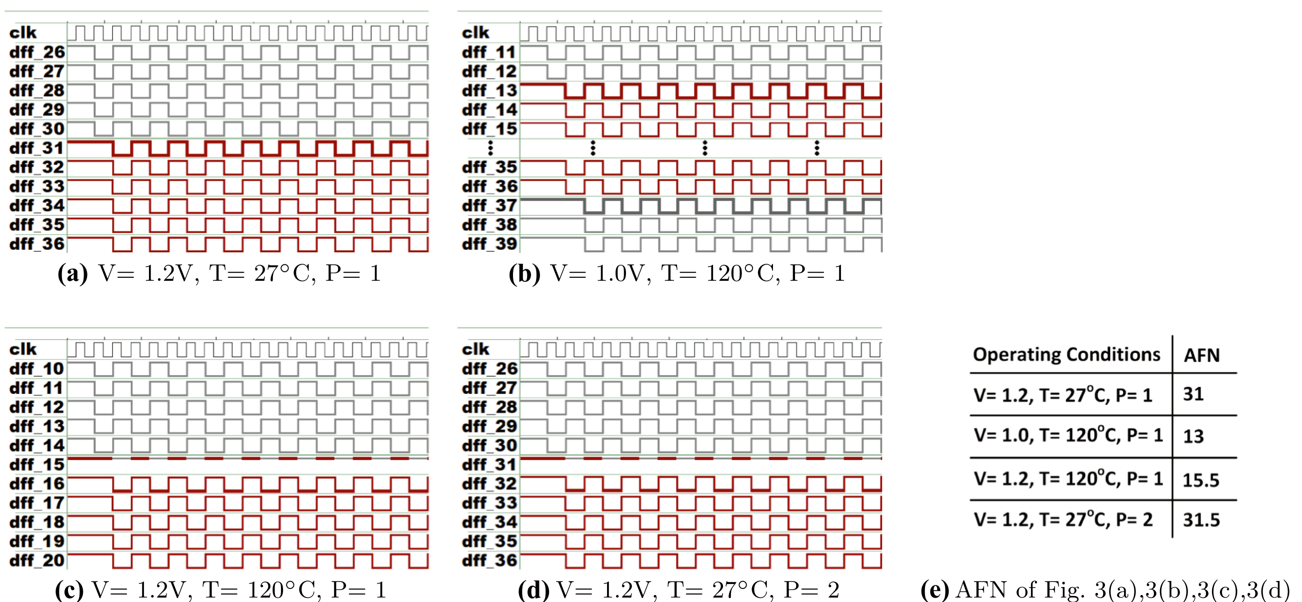
For the digital sensor shown in Fig. 2, sample waveforms for various combinations of voltage, temperature and process variations are depicted in Fig. 3 where the sensor consists of $n_0 = 9$ leading buffer followed by $n_1 = 43$ buffers each feeding one flip-flop. Figure 3e reports the AFN values related to all cases depicted in Fig. 3a–d. Note that when computing AFN we ignored the first 2 clock cycles as the circuit is not stable initially. As expected, temperature increase results in lower AFN (comparing Fig. 3a, c). Similarly, comparing Fig. 3b, c confirms our previous discussion on the increase of AFN in faster operating conditions (higher voltage). Finally, the AFN value is slightly affected by process variations; changing from 31 to 31.5 (in Fig. 3a, d) when operating under $V = 1.2V$ and $T = 27$°C for 2 different Monte Carlo simulations. Note that as shown in Fig. 3b, an operating condition may result in multiple phase changes in one clock cycle. In this figure, in each clock cycle the first change occurred in flip-flop indexed as 13 (so the AFN = 13) and the second change occurred in flip-flip indexed as 37.

The inference from the above observations is that AFN is a promising metric to reflect the environmental conditions under which a chip is operating. In fact, AFN can be used for implicitly detecting system's failure by checking if the operating conditions (e.g., temperature, voltage and clock frequency) are out of spec or not. To do so, we consider two thresholds for AFN so-called AFN_l and AFN_h. During the runtime, the current AFN is compared with these threshold values and based on the outcome of such comparison an alarm is raised or not (more details are given in Sect. 4.4). Accordingly, We assume that a sensor is embedded in the target chip along with the target circuitry, and its AFN value can be leveraged to represent the circuit's operating condition at runtime.

4.3 Trade-offs in AFN Calculation

The value of FN_i may not be identical in all clock cycles. For example in Fig. 3c, the first phase change occurs either in the flip-flop indexed as 15 or as 16, thus resulting in AFN = 15.5. In practice, the index of the flip-flop that observes the first phase-change in each clock cycle can be altered for different reasons as discussed below. First of all, during the runtime depending on the operation of the actual chip, the operating voltage and temperature of the chip may change, e.g. running a heavy code may raise the temperature in an accelerated manner. In that case, the FN_i can differ in the relevant clock cycle (CC). Secondly, the FN_i for a 0 propagation and a propagation of 1 might be different. On certain technological nodes, the propagation of 0 is quicker than the propagation of 1 across the buffer chain. Here the sensor can experience a lower FN_i when the delay chain is experiencing A) and a higher FN_i when the delay chain is experiencing $\overline{A}$ (say 1 → 0), because a discharging for 1 → 0 might be faster than charging for 0 → 1. Thirdly, various operations can increase noise in the system resulting in a varying FN_i. In addition, an attack launched on the chip can change the FN_i dramatically.

In the light of the above cases, it is necessary to gather the information in consecutive clock cycles. To do this, we need to collect the FN for a few CC and compute the average of the obtained values. The average of the considered FNs



(a) V= 1.2V, T= 27°C, P= 1 (b) V= 1.0V, T= 120°C, P= 1 (c) V= 1.2V, T= 120°C, P= 1 (d) V= 1.2V, T= 27°C, P= 2

Operating Conditions	AFN
V= 1.2, T= 27°C, P= 1	31
V= 1.0, T= 120°C, P= 1	13
V= 1.2, T= 120°C, P= 1	15.5
V= 1.2, T= 27°C, P= 2	31.5

(e) AFN of Fig. 3(a),3(b),3(c),3(d)

Fig. 3 Waveforms of Fig. 2 in different operating conditions. Voltage, Temperature and Process are displayed with their initial letter V, T, and P, respectively

is then used in the digital sensor characterization discussed in Sect. 4.2. However, the number of clock cycles that we consider during the AFN measurement is an important factor in the precision of our digital sensor. The basic principle is that the higher the amount of CCs considered in calculating AFN, the more details we obtain during the AFN measurement, thus making the sensor characterization more stable during the operation of the chip. However, the hardware overhead will increase by averaging more *FN* values (will be discussed in details in Sect. 6). In addition, if the number of CCs considered for calculating AFN is too high, some timing-based attacks or environmental modifications that occur in only a few CCs can be skipped.

As mentioned earlier, for AFN calculation we gather the FN_i values and find their average. In order to do this, we require an accumulator and a divider. As a divider can impose a lot of hardware overhead, we suggest observing a power of 2 CCs . This lifts the need for a divider circuitry as in this case the sum of FN_is is simply shifted based on the number of considered clock cycles to find the AFN value. In our simulation, we considered the cases of 2, 4, and 8 for CC_i.

As FN_i value may change slightly by noise, averaging these values gives a more stable characterization metric to be used for raising an alarm if needed. Indeed by such averaging, the variance of FN_is due to noise is eliminated. Figure 4 depicts a sample waveform related to the case in which operating voltage has been changed at runtime. This figure has been extracted for the fixed temperature of 27°C where the voltage changed in different points of time. As shown, for clock cycles 3 to 5 (CC_{3-5}), *FN*s are 31, The *FN*s change to 30 for CC_{6-7}. Finally, it changes to 31, 32, 31, and 32 in the following clock cycles for voltage 1.20V, 1.25V, 1.20V and 1.25V, respectively. This demonstrates the need to eliminate the noise effect during runtime by averaging FN_is, and confirms the need for using AFN for sensor characterization. Note that voltage variation is presumably much slower in practice, owing to P/G network capacitive load. However, in case of attack, the voltage may change gradually or abruptly by the adversary.

As mentioned, averaging FN_is under a high number of clock cycles may result in skipping an attack without detecting it or in environmental manipulation. For example, if the FN_i is decreased due to a fault or an attack, higher FN_is in the next clock cycles can compensate for it and the AFN value may still remain in range. Accordingly, the chances of such exploitations increase if the number of clock cycles under which AFN is calculated is increased. Therefore, the number of CCs must be carefully chosen based on the application. Moreover as will be discussed in Sect. 6, our sensor generates an *Error* signal when the difference of two consecutive FN values (computed during runtime) exceeds a predefined threshold. This signal can also help to detect

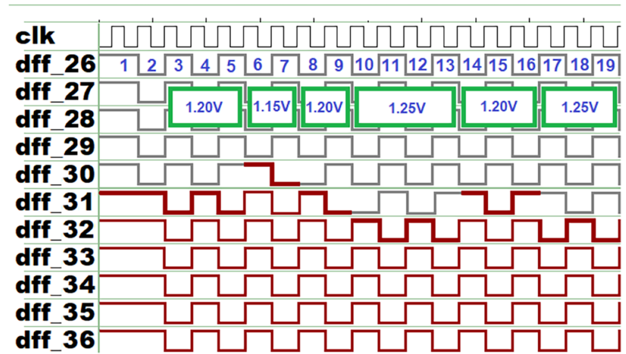


Fig. 4 Waveform of digital sensor with varying voltage and fixed T= 27°C

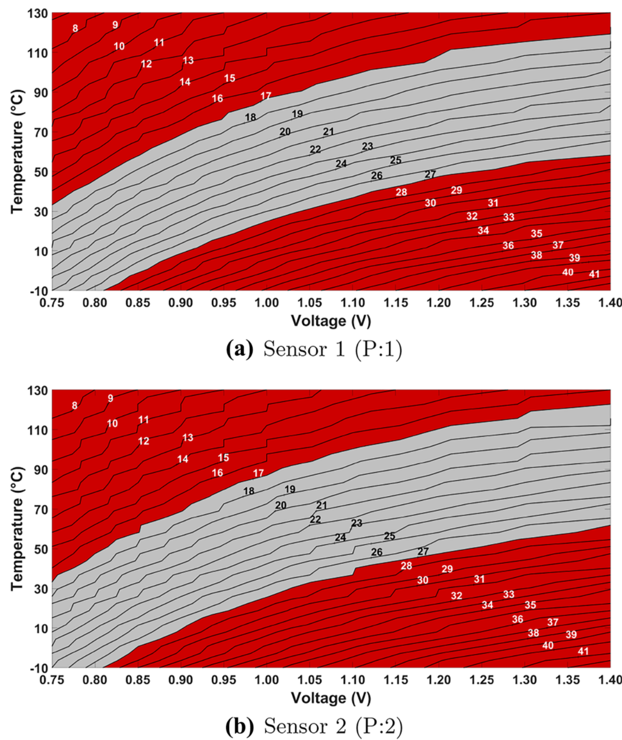
an abrupt changes in environmental conditions; thus giving hints about possible attacks.

4.4 Raising Alarm Based on Operating Conditions

As mentioned earlier, in this research, we deploy the sensor's AFN to predict whether the system operates out of specification and an alarm is raised if so. During the runtime the AFN quantity is calculated in each clock cycle and this value which represents the current operating condition is compared with the lowest and highest acceptable AFN values (AFN_l and AFN_h) that have been decided by the designer based on the spec referring to the worst and best case conditions.

The alarm will notify the user when the circuit is operating slower or faster than the nominal situations. In this paper, we consider the nominal case as $Vdd = 1.2V$ and temperature 75°C. Our simulations (in Fig. 5) show that in this operating condition, for our sensor shown in Fig. 2 AFN=22. Then we consider a trust spectrum around this AFN to ignore negligible changes in operating conditions and measurement noise; we call it the confidence range. In this paper, without loss of generality, we consider the confidence range of [-5,+5], i.e., any measured AFN between 22±5 is considered as acceptable, while an alarm is fired otherwise. Here we refer to 17 as the AFN_l and 27 as the AFN_h.

In practice, it is necessary to recognize a slower state (the conditions with AFN values lower than the accepted range) since the system's timing criterion is not fulfilled in slower situations. Thus, the device would sample a wrong value during an operation and it may induce ultimate failure. In comparison, faster conditions (denoted with AFN values higher than the accepted range) can also result in incorrect sampling generating failures in some applications (e.g., when the attacker increases the internal clock frequency or inserts a clock glitch in case of fault injection attacks.)



(a) Sensor 1 (P:1)

(b) Sensor 2 (P:2)

Fig. 5 Contour graphs depicting AFN variations in different PVT conditions for two fresh sensors (age:0) both realized from the same design while experiencing different process variations. The parts shown in red depict the operating conditions under which an alarm is raised, while the grey area displays the conditions considered as safe

Figure 5a illustrates the AFN in different voltage and temperature combinations in our sensor. As expected, AFN is lower for the conditions in which the underlying circuit operates slower, i.e., in low voltages and high temperatures, while its value increases by moving towards lower temperatures and higher voltages. In this figure, the parts shown in red depict the operating conditions under which an alarm is raised, while the grey area displays the conditions considered as safe (based on the confidence interval we discussed earlier). Note that the upper red part denotes to the conditions in which circuit operates slower, while the lower red portion points to the conditions in which circuit operates faster than expected. For example, as depicted when $Vdd = 1.0V$ and $Temperature$= 100°C, AFN is 15, thus an alarm is raised due to a slow operation ($AFN < 17$). Similarly, under $Vdd = 1.3V$ and $Temperature$= 30°C, AFN is 31. Here an alarm is fired due to operating faster than expected ($AFN > 27$). In contrast, when $Vdd = 1.2V$ and $Temperature$= 80°C, no alarm is raised as AFN is 20. To show the impact of process variations, Fig. 5b presents the AFN values for the same sensor design, realized via Monte-Carlo simulations in HSpice. Comparing this figure with Fig. 5a confirms that process variations have a slight impact on the AFN value extracted in each (V,T) pair.

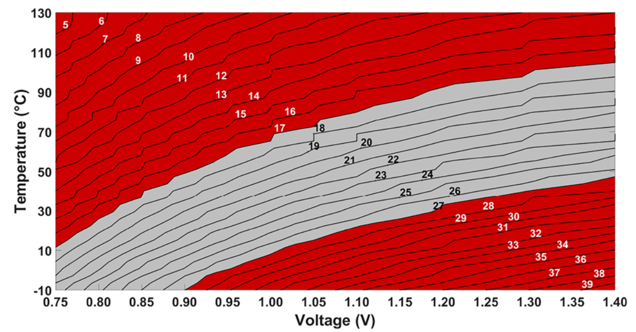


Fig. 6 Contour graphs depicting AFN variations in the 6-month old Sensor 1 (recall Fig. 5a)

4.5 Impact of Aging on Raising Alarms

As mentioned earlier, aging results in the increase of the threshold voltage of the underlying transistors, and makes the related gates slower over time. This results in inaccuracies in raising alarms as the AFN quantity that is extracted in a (V,T) pair may change due to aging. Thus comparing the runtime AFN with the preconsidered AFN_l and AFN_h (i.e., the confidence interval) may result in missing an alarm or raising an alarm when not necessary thus jeopardizing the security or availability of the chip.

To depict the aging impacts on AFN values, Fig. 6 displays the AFN quantities for the same sensor used throughout of this study in each (V,T) combination when the sensor has been used for 6 months. Comparing this figure with Fig. 5a (the fresh counterpart) shows that in some operating conditions the aged version fires an alarm while the new one does not, e.g., in case of $Vdd = 1.2V$ and $Temperature$= 90°C the AFN was 18.5 for the new device while decreased to 16.5 in the 6-month old device. This is a false alarm and should be avoided. On the other hand, for example, when $Vdd = 1.3V$ and $Temperature$= 40°C, the AFN was 28 for the new sensor (an alarm is raised) but decreased to 26.5 in course of 6 months which doesn't fire any alarm. Such missed alarm situations should be also lifted.

Comparing Figs. 5a and 6 shows that due to aging, the AFN confidence interval has shifted to the right. Indeed, for the operating conditions with an AFN close to but higher than AFN_l (17 in our case) in the new device, aging shifted the AFN from the gray area to the red area thus resulted in a false alarm. Similarly, for the cases close to the AFN_h border but with a higher value, due to the aging the alarm is missed, i.e., it is not raised. Figure 7a, b depict the (V,T) pairs that result in a false or missed alarm after aging for 6 months and 1 year, respectively. This shows that the rate of false and missed alarms increases during the digital sensor's lifespan. This calls for proper actions. Section 5 presents our proposed schemes to diminish the false and missed alarm rates.

To have a better observation of the aging-induced false and missed alarm rates, Fig. 8 depicts the percentage of missed and false alarm rates over the course of 7 years in every 2 months of usage. As shown, both false and missed alarm rates increase with aging. These rates are higher at the beginning and tend to saturate after some time of usage. In particular, the false alarm rate increases to 21.88% in the first 2 years, while 36.93% in 7 years. Similarly, 15% of alarms are missed after 2 years of usage and 22.85% are missed after 7 years. These findings confirm that we need a tuning or calibration system that will reduce the false and missed alarm rates, ensuring reliable alarm generation over the device lifespan.

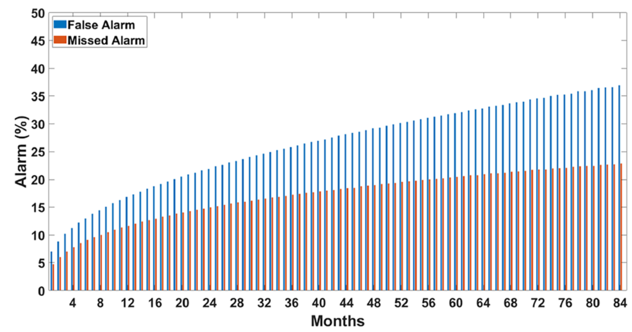
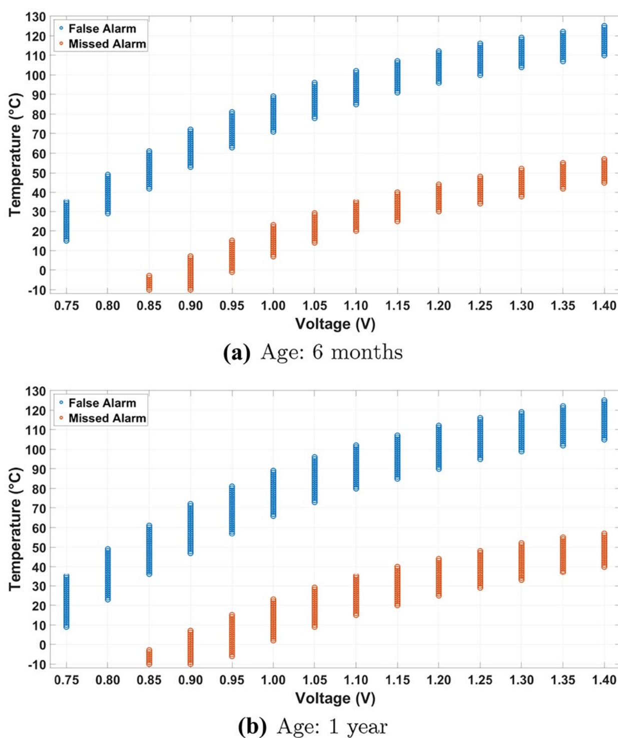


Fig. 8 Effect of aging on missed and false alarms

5 Sensor Calibration

As discussed earlier, the rate of false and missed alarms increases when the sensor is aged. This results in inaccuracies in sensing environmental conditions and jeopardizes the availability and security of the system in which the sensor is embedded. One may think of characterizing the aging effects on the sensor and tabulate it for calibrating the AFN value based on which alarm raising decision is made. However, firstly, embedding such a memory in our lightweight sensor is too much costly in terms of hardware overhead and, secondly, we need to have information about the time during which the sensor was ON (thus aged), as well as the exact operating condition during that time (i.e., the exact value of voltage and temperature). Accordingly, this solution is not feasible for the designed sensor.

In order to have an accurate and reliable value of AFN during the course of sensor's lifetime, we propose to duplicate the embedded sensor. The sensor and its duplicated counterpart both have the same design yet their specifications may be slightly different due to process variations. Note that both sensors are placed next to each other so they sense the same temperature and both are fed with the same voltage. We use the second sensor to calibrate the AFN value extracted from the other sensor during the runtime. In this section we propose two methods for such calibration, namely *Differential Calibration* and *Machine Learning Based Calibration*.



(a) Age: 6 months

(b) Age: 1 year

Fig. 7 Missed and false alarms in different aging conditions

5.1 Proposed Run-Time Calibration Schemes

5.1.1 Differential Calibration

The idea behind the Differential Calibration (DC) is simple but effective. As mentioned we embed two similar sensors in the chip. The second sensor is rarely turned on (we call it Rarely-on Sensor or R-Sensor). Indeed, it is always OFF except in the points of time that we need to extract the calibration values. We refer to these points of time as Calibration Time (TC) hereafter. Note that even at those points of time, the R-Sensor gets ON for a short time so it doesn't get aged (for example it would be ON one time per month only for a couple of clock cycles). Thereby, its AFN value (AFN_R) would be similar to the AFN value it would represent when it was new under the same (V,T) combination. Indeed in this method, we leverage the AFN extracted from the second sensor (AFN_R) at time TC_i (when it was ON) to calibrate the first sensor (Always-on Sensor or A-Sensor) whenever the R-Sensor is not accessible, i.e., the R-Sensor is OFF.

Note that the R-Sensor may get ON in a periodic manner or ad hoc (albeit rarely). Also, there is a possibility that both sensors are OFF simultaneously for some time, e.g., when the system in which the sensors are embedded is unplugged from power supply. This does not affect the efficacy of the proposed methodology, and in all such possible scenarios, our proposed calibration method works well.

In practice, we need to calibrate the extracted AFN of the main sensor (A-Sensor) in each (V,T) condition such that it represents the AFN for that (V,T) condition when the sensor was new. To do so, we need to know the amount that the A-Sensor's AFN decreased due to aging to compensate for such a difference. We use the R-Sensor's AFN each time it gets ON to find that difference approximately. As the R-Sensor was rarely ON, its AFN represents the AFN of a new sensor. Although the value of AFN_R-AFN_A is not exactly similar in all (V,T) combinations, our experimental results show that its variance across different (V,T) pairs are not significant. Thus, such a difference can be found any time that the R-Sensor is ON (TC_i) to be used to calibrate the A-Sensor's AFNs in the time duration between TC_i and TC_{i+1} where TC_i refers to the ith time the R-Sensor gets ON during its lifetime.

It is noteworthy to mention that the value of AFN_A gets lower and lower over time, albeit in the same (V,T) combination, as aging makes the circuit slower. Thereby AFN_A needs to be calibrated as mentioned to represent a fresh sensor. The experimental results shown in Sect. 7 confirm the efficiency of the DC scheme in decreasing the rate of false and missed alarms over time.

Figure 9a shows the overview of the process performed in calibration time of TC_i. Here, the calibration time refers to the time when both sensors are ON. In the calibration time the AFN value of both sensors are extracted and their difference is calculated (as shown in Eq. 1). The chip controller should configure the Adder/Subtractor mode in subtract mode to find this difference (δ (in Fig. 2). On the other hand, during the runtime so called operating time, the δ value is used to calibrate the AFN_A, and finding Calibrated-AFN_A (C-AFN_A) whose value is used by the system to decide if an alarm should be raised or not. As shown in Fig. 9b, the value stored in the memory is added to the AFN_A at any point of time between $[TC_i, TC_{i+1}]$. At TC_{i+1} the R-Sensor gets ON again and the δ value is updated. Thus, in the operating time, The chip controller should configure the Adder/Subtractor to the addition mode.

$$\delta = AFN_R - AFN_A \tag{1}$$

Note that in this paper the calibration process is performed in hardware (using one Adder/Subtractor and a memory element), however it is possible to conduct it in the software level if there is supervisory software with an interface with chip. In both cases, it is clear that the imposed cost is negligible.

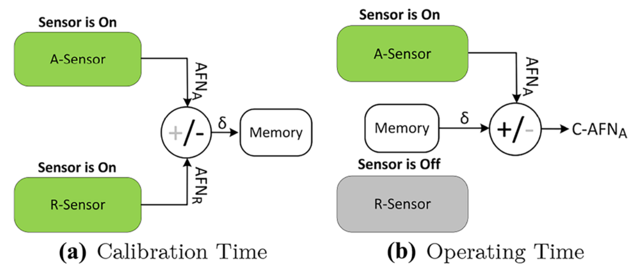


Fig. 9 Differential Calibration Method. C-AFN_A is used to decide if an alarm should be raised or not. In the calibration time both sensors are ON and in the operating time only A-Sensor is ON

5.1.2 Machine Learning Based Calibration

To diminish the rate of aging-induced false and missed alarms even more, we augment our DC method with a Machine Learning (ML) scheme to have a more accurate calibration during the runtime. We use a supervised machine learning scheme, with calibrated AFN as its labels and previous readings of AFN as its feature. Indeed, the concept of calibration is compatible with the regression problem in machine learning as we have an almost continuous change of AFN due to the operating condition variations (albeit in absence of attacks). Thereby, in the Machine Learning Based Calibration (ML-DC), among available ML schemes such as Neural Network (NN), Support Vector Regression (SVR), Linear Regression (LR), etc. We selected LR to compensate the aging-induced AFN changes as LR is lightweight. This results in much lower overhead if we want to implement it on hardware, and also it does not have any hyperparameter for tuning (in standard version as we used in this paper).

By using LR, a relation between current AFN_A, and a few previous readings of AFN_A as well as AFN_R at the time points when both sensors were ON simultaneously is found. This relation is presented with a first-degree polynomial whose coefficients are tuned during the training phase conducted beforehand as discussed below. The first degree polynomial has been considered as the aging impact is almost linear in average after a few weeks of aging. The results presented in Sect. 7 confirm this hypothesis. The model is used during the runtime to infer the current aging-compensated AFN_A.

Figure 10 shows an overview of the ML-DC approach in the calibration time. In this method, we add a small buffer (more precisely a shift register) in the chip to always store the last $M-1$ readings of AFN_A and AFN_R quantities when

both sensors were ON (related to the TC_{i-M+1} to TC_{i-1} time points). Our experimental results show that with an M as small as 5, we can get high accuracy. Then at time TC_i when again both sensors are ON, the current value of AFN_A along with the $2M-2$ values stored in the shift register are used to infer the aging-compensated AFN_A which we call AFN'_A. Ideally, this value should be the same as AFN_R at the same point of time but due to the ML approximating nature as well as the effect of process variations, they can be slightly different. We find their difference (δ) as shown in Fig. 10 and use this value during the operating time when only A-Sensor is ON in order to calibrate its extracted AFN (i.e., AFN_A).

Similarly, during the operating time (i.e., runtime), when only A-Sensor is ON, the current value of AFN_A along with the values stored in the shift register are used to assess the AFN'_A which is added up with the δ value computed at time TC_i in order to find the calibrated AFN_A (i.e., C-AFN_A) based on which alarm rising decision is made.

Training the model is done offline in software and then the model is embedded in the chip that includes the sensors. As we use LR for training, the model is low cost and the hardware overhead is not significant. The details on overhead are discussed in Sect. 7. The training data can be either obtained from simulation or from the fabricated chips. In the former case, we use the simulation data of K sensor pairs when they are new as well as when they are aged for different aging durations. This approach has two advantages. Firstly, training the model using the data related to multiple sensor pairs results in mitigating (or if the dataset is large enough removing) the effect of process variations in the sensor's outcome. This is due to the fact that the model learns the process variation effects gradually during the training with multiple sensor pair data, and benefits from such learning in inferring δ value which in turn used for finding the C-AFN_A. Secondly, conducting Monte Carlo simulations relieves us from the need for

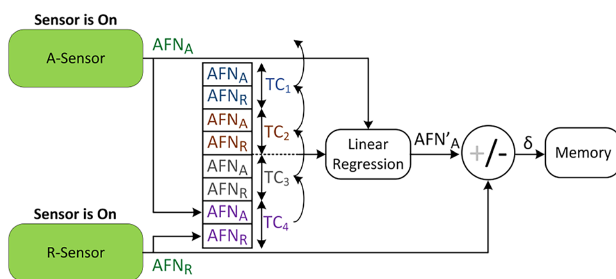


Fig. 10 Deploying the ML-DC method to extract the δ value in the calibration time when both sensors are ON to be used for updating the AFN_A value during the operating time when only A-Sensor is ON. DC is applied on top of LR. Here $M = 5$

multiple fabricated-chips data. To do so, we can conduct Monte-Carlo aging simulations (e.g., via HSpice) in different voltage and temperature conditions using aging aware technology libraries. The second option is scarifying a few chips to extract the data needed for training the model. In this case, we use K (K can be as low as 1 as our results confirm) chips after fabrication and place them in a climate chamber to accelerate aging under high temperature and voltage. We setup the chip such that the A-Sensor is ON and the R-Sensor is OFF when the chip is under stress. In regular time intervals, we make both sensors ON and perform AFN measurements. Such gathered data can be used for training the LR model.

Note that we consider a constant time interval between the calibration times (the time that both sensors get ON) for training the model. However, the time intervals between the calibration times can be variant during the chip usage. This is a realistic assumption as the data gathering (for training the model) is not performed during the chip usage and so it is completely under the control of designer (in case of simulation) or the fabrication facility (in case of using chips' data as mentioned above). However, during the chip-usage the user may decide to turn off the whole system, and in turn both sensors at some point in time. Therefore, the calibration may not be performed in fixed time intervals. The results discussed in Sect. 7 have been extracted with such an assumption and confirm the robustness of the proposed scheme even in such conditions.

5.1.3 DVFS Management

DVFS is very often implemented in modern chips because it allows to control at software level the power vs performance tradeoff, which plays an important role in User Experience. In practice, application specific chips only feature few DVFS valid configurations, such as 2 or 3 (e.g., fast & power-hungry, slow and power-efficient, etc.). Notice that more complex chips that allow to configure the DVFS in an arbitrary manner become vulnerable to PlunderVolt and sibling attacks [9, 10, 15–17]. We thus assume that the system limits the number of DVFS conditions to a small number. The digital sensor shall adapt to those conditions. In practice, each of those conditions is decided on the basis that the system remains equally safe. Therefore the thresholds need only be adapted, albeit marginally. This can be implemented through a reconfiguration of the digital sensor threshold saved in a given non-volatile (and immutable) memory. To conclude, our perturbation detection method with digital sensors can apply equally well in various DVFS conditions, provided there is a means to carry out small reconfigurations upon DVFS changes.

6 Hardware Implementation of the AFN Calculator and Alarm Generation Circuitries

Figure 11 depicts the hardware implementation of the circuitry used to assess the AFN value of each of the R-sensor and A-sensor in each clock cycle. In this figure, our digital sensor (recall Fig. 2) is shown in green. Let say it is the A-Sensor. In each clock cycle (CC_i), the values stored in the sensors' embedded flip-flops (i.e., $O_1, O_2, \ldots, O_{n_1}$) are given to the "*Position Detector*" circuitry in order to extract the index of the first flip-flop that experiences a phase shift to be used for sensor characterization (recall Sect. 4.2). The "Position Detector" can be as simple as a priority encoder. This index (FN_i) is an integer value in $[1, n_1]$ where n_1 refers to the number of flip-flops the sensor is composed of. The averaging of FN_i over $N = 2^{SEL}$ clock cycles evaluates the AFN value where *SEL* is a primary input to the sensor circuitry. This architecture is also used to detect an "error" that corresponds to a significant change (called *THR*) between two consecutive FN_i. This is necessary because the AFN is an averaged number, so it can filter out abnormal cases when there is a short duration disturbance.

The *AFN Calculator* module evaluates the AFN quantity based on the values of the FN_i in the last $N = 2^{SEL}$ clock cycles stored in a buffer (in particular shift register) in this module. Indeed such shift register keeps the last *N* values of *FN*s and its content is updated in every clock cycle when a new FN value is shifted in. Without loss of generality, Fig. 12 shows this implementation for the case where *SEL* is a 2-bit input. Here, we can configure the hardware during the runtime to decide about the number of clock cycles based on which we want to take the average and compute AFN, i.e., based on the application, we may want to compute AFN for the last *N* clock cycles (where $N \in \{1, 2, 4, 8\}$ in Fig. 12). This can be controlled by the "*SEL*" signal. In each clock cycle, the "*SUM*" signal represents the summations of all FN_is during the last *N* cycles which are used to find the AFN.

In this implementation, averaging is as simple as shifting the *SUM* value to the right. As shown, *SUM* keeps the sum of the last *N* readings of FN_i and is updated in each clock cycle *i* by adding the last FN_i value and subtracting the oldest one (FN_{i-N}). Thereby our run-time AFN calculation circuitry overhead is small and the AFN is calculated in every clock cycle. Indeed *SEL* is given as a primary input to the sensor circuitry (Fig. 11) and $N = 2^{SEL}$ shows the number of clock cycles used in AFN calculation. This enables the user to decide the number of clock cycles for AFN calculation during the run-time. Note that *SEL* and thus *N* should be fixed during the operation, and changing its value requires resetting the AFN calculation circuitry. The AFN value is not valid in the first *N* clock cycles after resetting.

Note that the shift register shown in Fig. 12 gets reset when the *Reset* input signal is asserted. Moreover, in each clock cycle, the calculated AFN value is extracted, and then calibrated using either the DC or ML-DC schemes (discussed earlier) to be used in each clock cycle to make decisions on raising alarms based on Eq. (2) where as discussed earlier AFN_l and AFN_h refers to the lowest and highest acceptable AFN values, respectively. As discussed earlier, AFN_l and AFN_h are decided based on the acceptable range of operation for each application.

$$Alarm = \begin{cases} \text{"0"} & \text{when } AFN_l \leq AFN \leq AFN_h \\ \text{"1"} & \text{otherwise.} \end{cases} \quad (2)$$

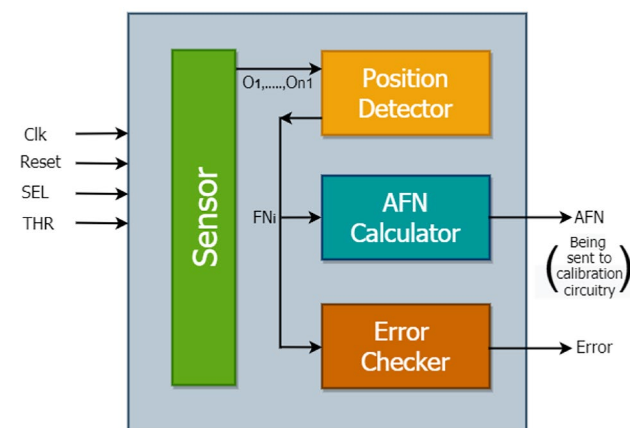


Fig. 11 Hardware implementation of sensor peripherals

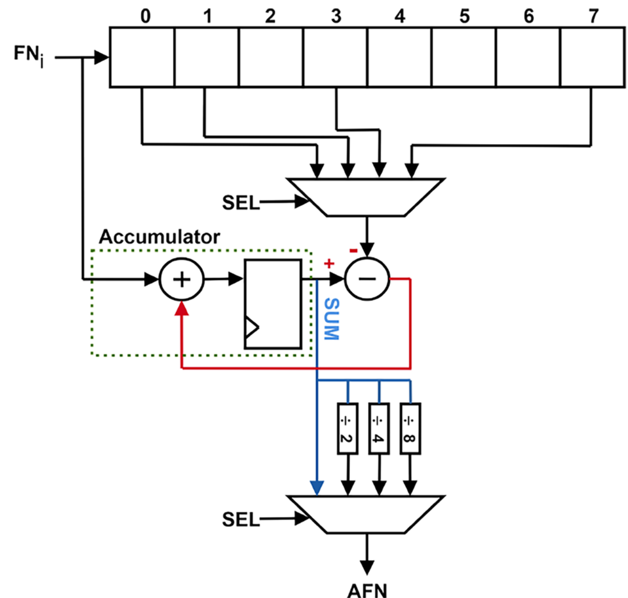


Fig. 12 Hardware implementation for AFN calculation. The AFN calculator is shown for the case where *SEL* is 2 bit (thus $0 \leq SEL \leq 3$)

The *Error Checker* block in Fig. 11 looks for hard errors occurring during the runtime or the attacks that drastically change the FN_i value from one clock cycle to the following one. Indeed if the difference of FN_i and FN_{i-1} is higher than a predefined threshold *THR*, an error signal is asserted to show the abrupt change of the sensor's outcome (Eq. 3). Such an error can be due to an intentional or unintentional voltage glitch or clock glitch. The value considered for *THR* depends on the application.

$$\triangle FN = |FN_i - FN_{i-1}|$$
$$Error = \begin{cases} \text{“1”} & \text{when} \triangle FN > THR, \\ \text{“0”} & \text{otherwise.} \end{cases} \quad (3)$$

7 Experimental Setup and Results

The sensor circuitries we implemented in this research include n_0=9 leading buffers followed by n_1=43 buffers and flip-flops (refer to Fig. 2). This sizing has been determined in order to have at least one phase change for all the PVT corners, for the range of (V, T) we considered in this study. The sensors were implemented at the transistor level using 45 nm NANGATE technology [39]. For transistor-level simulations, we used Synopsys HSpice and the built-in HSpice MOSRA Level 3 model to evaluate the impact of NBTI and HCI aging. The sensors' output were extracted under different voltage and temperatures and for different aging durations; up to 7 years of operation in steps of one month. The sensors were simulated for temperatures between −10°C and 130°C with 1°C steps, and for the voltage source (*Vdd*) betwee n 0.75V and 1.4V with 0.05V steps. We realized 5 different sensor pairs using Monte Carlo simulations to evaluate the efficiency of the proposed schemes in different process variations. The DC method is not affected by process variations as in this method each sensor pair uses its own data for calibration. Note that the effect of process variations in age=0 (when the device is new) results in a δ value that is used for calibration of A-Sensor in all of its readings till the next point of time when the R-Sensor is also ON (after 1 month in our experiments).

To investigate the impact of process variations for the ML-DC scheme, we trained our model (linear regression) with the AFN values from "one" sensor pair (A-Sensor and R-Sensor) and used the model to infer the AFN values for another sensor pair. The results are promising confirming the negligible impact of process variations in the ML-DC scheme. For our investigations, we conducted 5 Monte Carlo simulations using a Gaussian distribution: transistor gate length L: $3\sigma = 10\%$, threshold voltage V_{TH}: $3\sigma = 30\%$, and gate-oxide thickness t_{OX}: $3\sigma = 3\%$. As mentioned earlier for ML-DC we used a linear regression scheme due to its low cost and at the same time high precision in our case.

7.1 Impact of Aging in the Sensor's Characterization

The first set of results depicts the impact of aging on the AFN value extracted for different (V,T) combinations. Recall that AFN quantity is used for characterizing the target sensor. The graphs shown in Fig. 13 depict the AFN values in each considered (V,T) after 1, 2, 3, and 4 years of aging. These results have been extracted for the same sensor whose AFN values in different (V,T) combinations when the sensor was fresh (not-aged) were shown previously in Fig. 5a.

Comparing Fig. 5a with the graphs shown in Fig. 13 confirms the impact of aging on the AFN values. In particular, as expected the device gets slower with aging and thus the setup time violations occur in the flip-flops in lower indexes when it is aged, thus resulting in lower AFN values. Such shifts can be clearly observed in the graphs shown in Fig. 13. The higher the age, the more decrease of AFN value in the same operating condition.

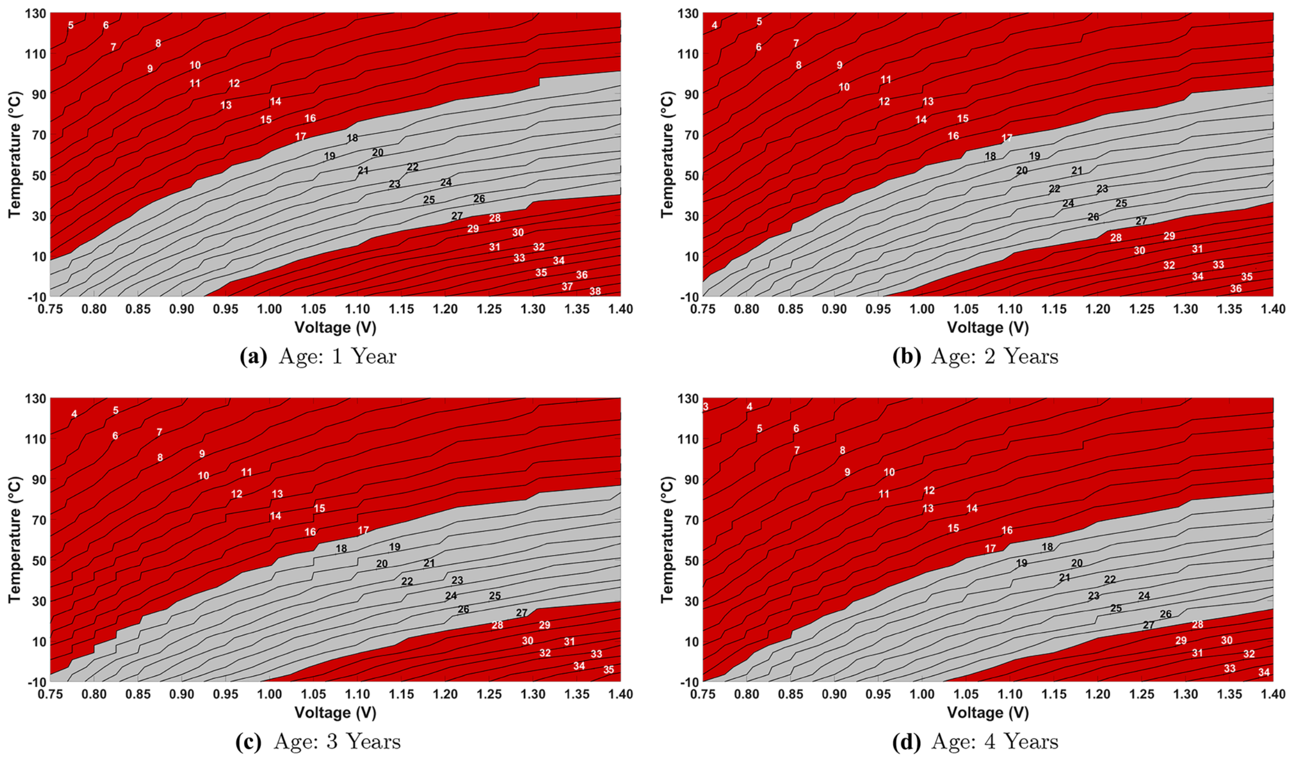
The takeaway point from these observations is that as the AFN value changes with aging, deciding about raising alarms solely based on AFN during the run time may result in inaccuracies in firing alarms. This confirms the need for an efficient calibration scheme to decrease such inaccuracies and justifies the calibration schemes proposed in this paper.

7.2 Missed and False Alarm Rates in the Sensors Equipped with DC and ML-DC Schemes

Differential based Calibration (DC): Figure 14 illustrates the false and missed alarm rates with and without DC calibration. In these experiments the R-Sensor is ON for 8 clock cycles per month and its AFN is used to calibrate the A-Sensor in the following month (till the next reading of R-Sensor). Thereby, the R-Sensor is not affected much by aging, and the rate of the aging-induced false and missed alarms in the R-Sensor is almost 0. When there is no calibration, we just have one sensor (i.e., the A-Sensor).

The results shown in Fig. 14a have been extracted for the first sensor pairs (lets say $A - Sensor_1$ and $R - Sensor_1$ in case of calibration and $A - Sensor_1$ only in the original case with no calibration). As depicted if the sensor is not equipped with DC calibration, the false alarm rate would be 17.5%, 23%, 30%, and 38% after 1 year, 2 years, 4 years, and 7 years of usage, respectively. Similarly, 13%, 16.5%, 21%, and 23.5% of alarms are missed after 1 year, 2 years, 4 years, and 7 years of aging, respectively when no AFN calibration scheme is deployed.

As observed in Fig. 14a the rate of false and missed alarms decreased tremendously when using DC. In particular, we can see that false alarms would be as little as 1% in some cases and less than 5% in almost most cases when using DC. Similarly, the missed alarm rate reduced to below 5% in most of the age instances. That is known to be

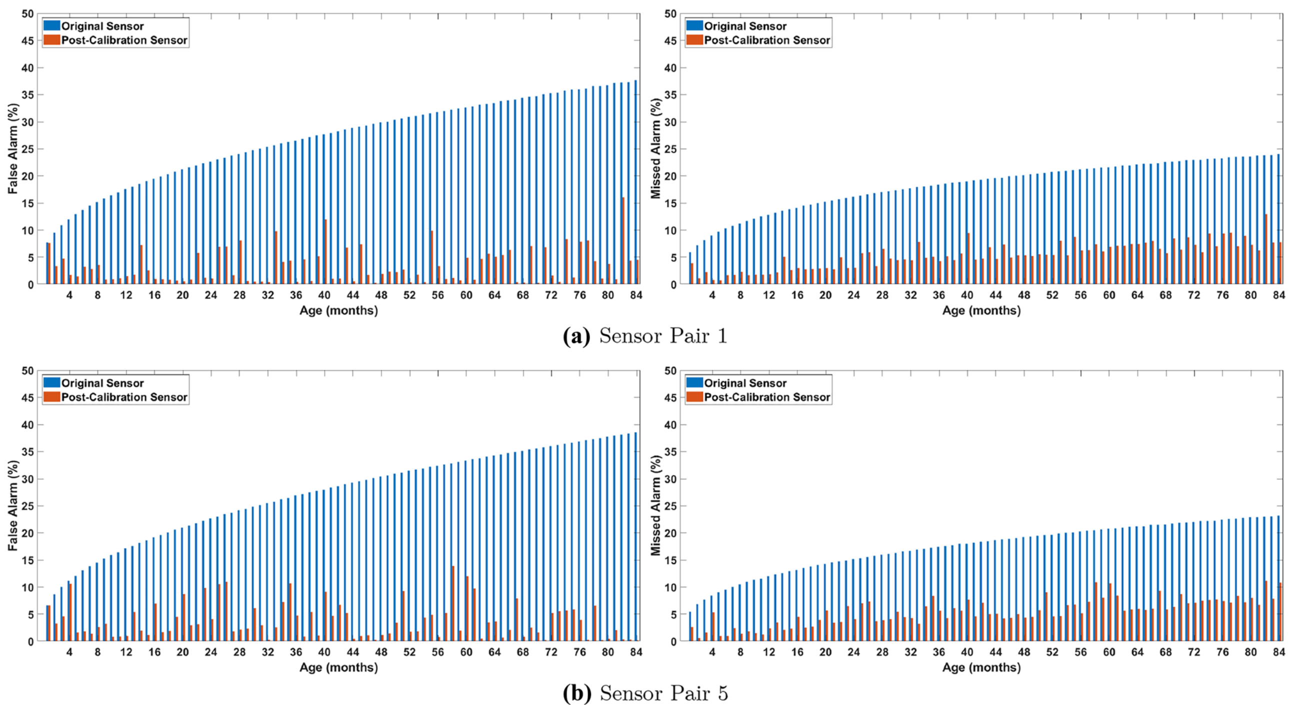


(a) Age: 1 Year **(b)** Age: 2 Years

(c) Age: 3 Years **(d)** Age: 4 Years

Fig. 13 Contour graphs depicting AFN variations in different aging conditions

a reliable edge case for the application of a sensor. Few picks are observed in missed and false alarm rates in Fig. 14a, e.g, at 33 or 40 months of aging in these experiments. This is because in the DC method, the R-Sensor turns on specific times (once per month in these experiments), and based on the (V,T) condition in that specific point of time, the



(a) Sensor Pair 1

(b) Sensor Pair 5

Fig. 14 False and missed alarms with and without sensor calibration via DC method

calibration value is decided. Thus such selection follows a random nature.

Figure 14b displays the false and missed alarm rates related to both an original and a post-calibrated digital sensor for another process variation ($A - Sensor_2$). Here, $R - Sensor_2$ was used for calibration. As shown in this figure, similar to the case shown in Fig. 14a, DC calibration tremendously enhances the accuracy of alarm generation, i.e. both false and missed alarm rates are reduced.

To depict the impact of process variations in more details, we performed DC calibration on 5 different sensor pairs realized from 5 Monte Carlo Simulations. Table 1 shows the average false and the missed alarms over the course of 7 years. As depicted, with DC calibration, the rate of false and missed alarms in $Sensor_1$ (referring to $A - Sensor_1$) drops to 3.30% and 5.33% from 26.70% and 18.13% in case of no calibration, respectively. The other sensors show a very similar result. For example, the $Sensor_5$'s false and missed alarm rates would decrease from 27.68% and 18.23% to 4.57% and 4.99% on average in the course of 7 years after DC calibration, respectively. Based on these results, almost all sensors experience the same rate of improvement in accurately raising alarms. we can conclude that using the DC calibration process, we were able to resolve 84.15% of the false alarms and 70.53% of the missed alarms. The takeaway point from these observations is that DC scheme works very efficiently regardless of process variations.

Machine Learning based Differential Calibration (ML-DC): Figure 15 illustrates how the false and missed alarm rates reduce when the ML-DC calibration scheme is deployed. In these experiments, as also mentioned earlier, the R-Sensor is ON for 8 clock cycles per month. To show that the impact of process variations is negligible in our results, we only used data from sensor pair 1 to train the linear regression model and tested the model for the other 4 sensor pairs.

For training the LR model, we randomly selected 1000 data points (in total). Each data point includes the AFN values of the $A - Sensor_4$ and $R - Sensor_4$ (in a randomly selected time TC_i which is between 1 and 84 months, and under a randomly selected voltage and temperature condition) along with the last 4 readings of the AFN values of these sensors when both sensors were ON (under other randomly selected (V,T) conditions). The model built during the training phase was then tested against all data points of the other 4 sensor pairs($> 165,000$ points related to different (V,T) combinations in different aging durations for each sensor).

Table 1 The average rate of false and missed alarms over the course of 7 years for 5 sensor pairs when DC is deployed and the R-Sensor is ON for a few clock cycles per month

Sensor Pair No.	False Alarm (%)		Missed Alarm (%)	
	Original	DC Calibrated	Original	DC Calibrated
1	26.70	3.30	18.13	5.33
2	27.77	4.14	17.58	5.19
3	26.96	3.77	17.25	5.39
4	28.48	5.13	17.75	5.30
5	27.68	4.57	18.23	4.99
Average	27.51	4.36	17.78	5.24

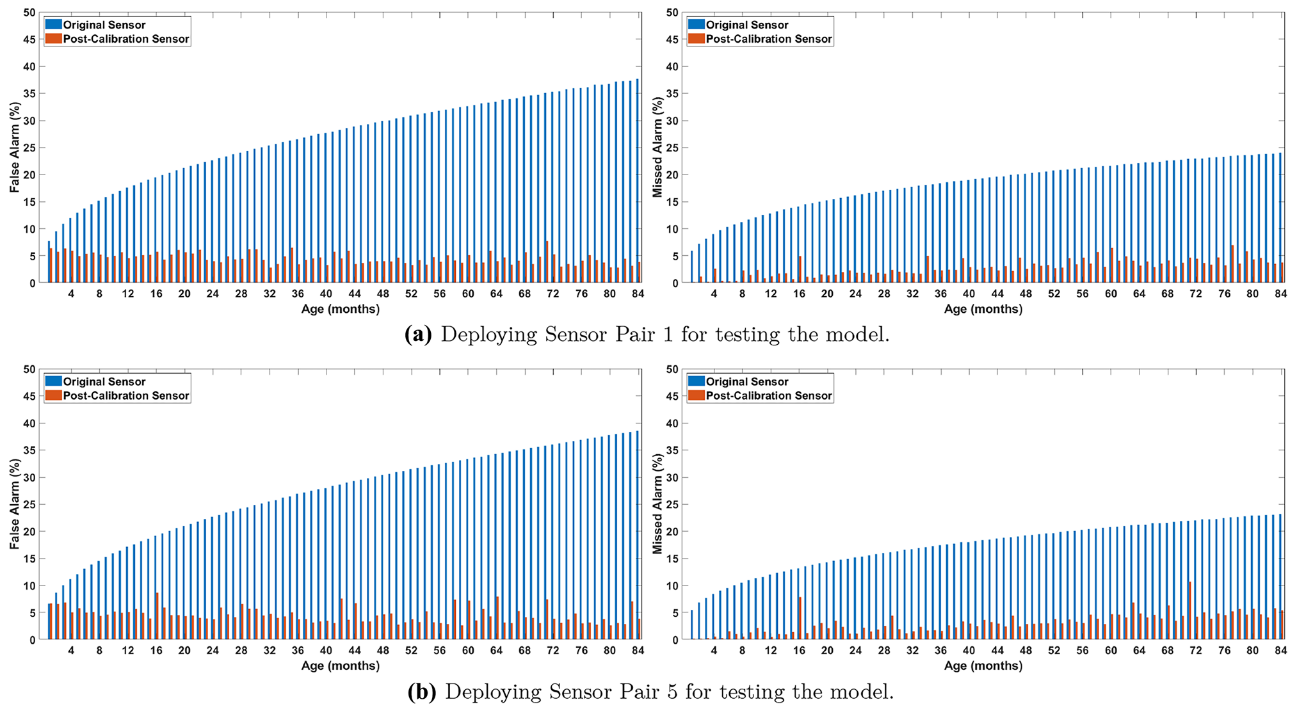
Note that in real applications the data used for training can be either attained via aging simulations (e.g., via HSpice) in different voltage and temperature conditions using aging aware technology libraries, or via scarifying only one chip after fabrication to use its data for training the model. The victim chip is placed in a climate chamber to accelerate aging under high temperature and voltage, and it is setup such that the A-Sensor is ON and the R-Sensor is OFF when the chip is under stress. Note that although a constant time interval between the calibration times (the time that both sensors get ON) is considered for training the model, the time intervals between the calibration times can be variant during the chip usage. The results shown for the ML-DC scheme are based on such assumptions.

Figure 15 depicts the false and missed alarm rates for Sensor pair 1 and Sensor pair 5 when ML-DC is used assuming that Sensor pair 4 was used for training the model. As shown in both cases miss alarm rate decreased more compared to the case when DC was used. Table 2 shows the effect of process variations in more details. Here, the average of the false and missed alarms over the course of 7 years are shown when the model was trained using the data from Sensor pair 4. As depicted, with ML-DC calibration on average, over the course of 7 years, the rate of false and missed alarms are 4.68% and 2.82%. Comparing these results with the results reported for DC in Table 1 shows that miss alarm rates decreased significantly ($\approx 2.42\%$ more). The false alarm rate in ML-DC is very slightly higher than DC (on average 4.68 compared to 4.36). Note that as mentioned also earlier, to ensure security missed alarms should be low. In practice, false alarms are more related to the availability of the device.

The takeaway point from these observations is that ML-DC can decrease the missed alarms considerably, thus is a promising solution when security is taken into account.

7.3 Effect of Process Variation on the Calibration Schemes

To illustrate the process variation effects more clearly, we present the distribution of the alarms raised by our sensor when no calibration scheme is deployed as well as when DC or ML-DC was used. The heatmaps depicted in Fig. 16 illustrate how many of the 4 target sensor pairs will raise



(a) Deploying Sensor Pair 1 for testing the model.

(b) Deploying Sensor Pair 5 for testing the model.

Fig. 15 False and missed alarms with and without calibration via ML-DC method. Sensor Pair 4 was considered as baseline for training in both cases

alarms in each (V, T) combination when the original sensor is new and no calibration scheme is used (Fig. 16a), when the original sensor has been aged for 42 months but no calibration scheme is used (Fig. 16b), as well as when DC or ML-DC were used for a 42 month old sensor (Fig. 16c, d). Since in ML-DC, one sensor pair is used for training and the other 4 sensor pairs are used for evaluation, for the sake of comparison we also show the results of 4 sensor pairs for the cases shown in Fig. 16a–c.

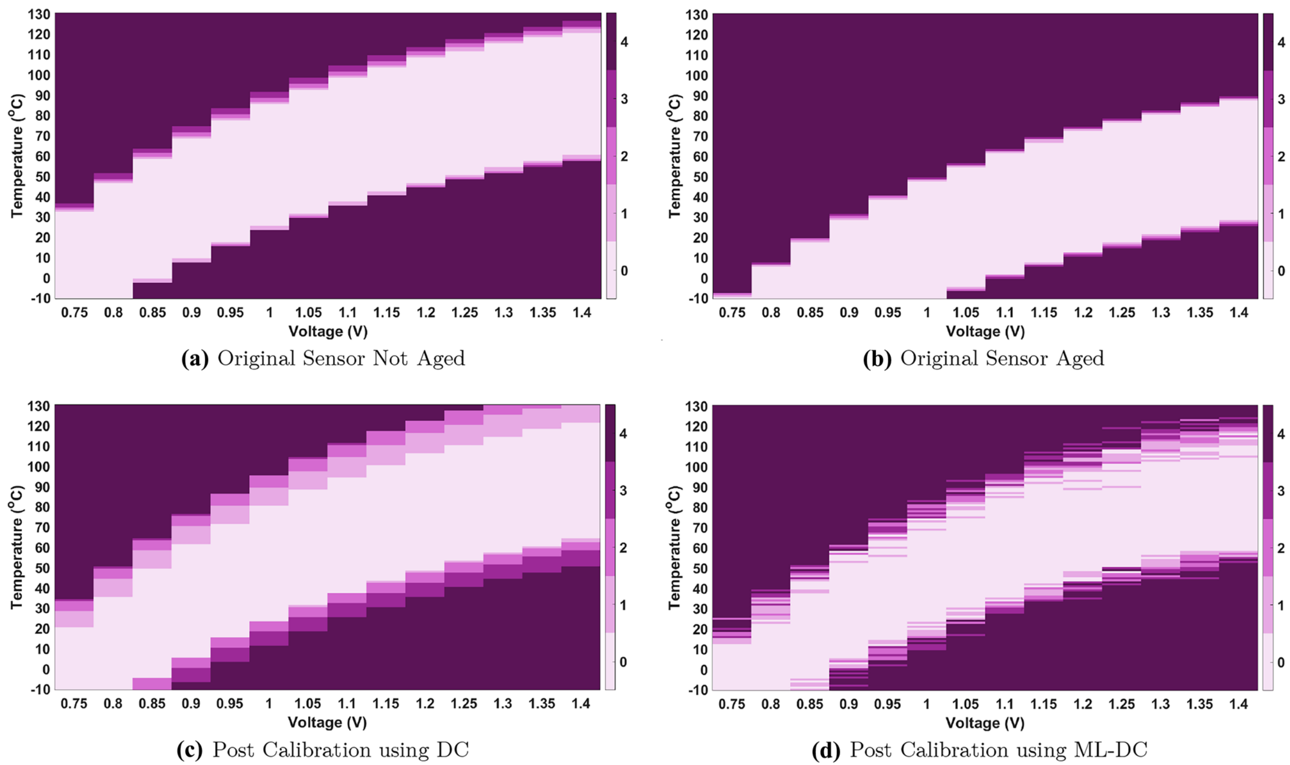
As shown in Fig. 16a, for the new sensors without calibration in 94.73% of conditions either no sensor raised an alarm or all of them raised an alarm. A very similar trend is observed when the device is aged for 42 months, i.e., in almost all cases either none or all of the sensors raise an alarm in a (V,T) combination. However, as shown and also discussed earlier, due to the aging the AFN values are changed (comparing Fig. 16b with Fig. 16a) and so some of the situations where none or all of the aged sensors raise an alarm, can be related to the cases where all missed raising an alarm or all raised a false alarm, respectively.

As depicted in Fig. 16c, d, when DC and ML-DC calibration is used respectively, again in most of (V,T) conditions either no or all sensors give alarm but comparing these figures with Fig. 16a shows that the rate of missed and false alarms has been reduced significantly when using these 2 calibration schemes. A few cases that result in 1-3 sensors raise an alarm is unavoidable due to process variations but as shown those cases are very infrequent.

The takeout point from these observations is that both calibration methods performed well on generating alarms in aged devices regardless of process variations as in almost all cases all 4 sensor pairs resulted in an alarm or none of them raised any alarm and the rate of missed and false alarms has been reduced significantly when these calibration schemes are used. Also, we observed that ML-DC was shown more efficient in missed alarms reduction, and in case of false alarm, DC and ML-DC are comparable, yet the reduction comparing to the original is substantial.

7.4 Overhead Imposed by DC & ML-DC Schemes

As mentioned earlier, the overhead of applying the proposed calibration techniques is insignificant. In particular, as depicted in Fig. 9, to realize DC, only one adder/subtractor and a memory cell is needed. On the other hand, based on Fig. 10, for ML-DC an adder/ subtractor, a memory cell and a shift register to keep the last N readings of each of the two sensors are needed. As our results shows with N as low as 4, ML-DC can considerably decrease the rate of false and missed alarms. In addition, in this case, we need 5 multipliers and 4 adders to implement the LR scheme in hardware. Note that the training is performed in software level and the



(a) Original Sensor Not Aged **(b)** Original Sensor Aged

(c) Post Calibration using DC **(d)** Post Calibration using ML-DC

Fig. 16 Effect of process variation on the raised alarms in 4 sensor pairs. The figures show how many of the 4 sensor pairs raised an alarm in each of (V,T) conditions

wights for the LR model are fed to the chip. For calibration, we duplicate the sensor (insert the R-Sensor). However, this replication is insignificant considering the fact that the sensors are mainly used in large systems and not small chips. As the sensors and in turn their calibration circuitries are not in the critical path of the main circuitry (they are separate), they impose no delay overhead to the circuit. Moreover, the power overhead is negligible as in the normal situation the environment is steady (no change in temperature or voltage or clock frequency). Thus the sensor circuitry does keep the same values (AFN) most of the time.

Table 2 The average rate of false and missed alarms over the course of 7 years for four sensor pairs when ML-DC is deployed and the R-Sensor is ON for a few clock cycles per month. Sensor pair 4 has been used for training the model

Sensor Pair No.	False Alarm(%)	Missed Alarm(%)
1	4.51	2.82
2	4.83	2.88
3	4.48	3.06
5	4.90	2.53
Average	4.68	2.82

7.5 Security of the Sensor Against Side-Channel Attacks on Protected Chips

It is clear that the data sensed by the raw digital sensors is very sensitive. Indeed, it allows for the chip self power monitoring. This is the premise of the PlunderVolt side-channel attack. As this attack definitely belongs to our threat model (recall Sect. 2.1), it shall be mitigated.

Clearly, sensors do bring sensitive information (a side-channel or even a subliminal communication channel). Therefore, environmental sensing shall be considered sensitive and shall not be outputted.

A same usage practice regarding sensors is therefore to restrict their usage to privileged processes, implementing the DVFS strategy. User processes are denied access to the sensors, since we have no means to distinguish a licit user from an attacker.

8 FPGA Implementation of the Deployed Digital Sensor

This set of experiments have been realized by implementing our sensor on a Xilinx Spartan 6 FPGA resided on a SAKURA-G board. The layout is shown in Fig. 17. We

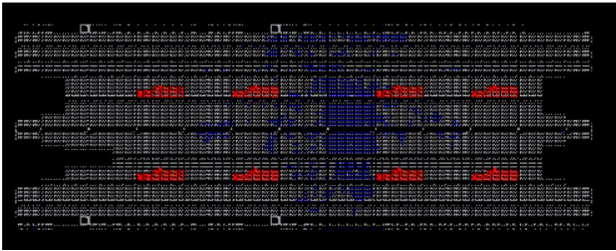


Fig. 17 The layout of the 8 sensors implemented on a single Spartan 6 FPGA resided on a SAKURA-G board

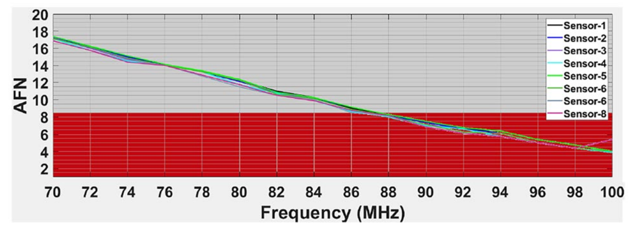


Fig. 18 AFN variations for 8 sensors implemented in an FPGA fabric when the circuit operates at different clock frequencies

implemented 8 sensors, shown in red, in 2 rows (on a single FPGA) each with 22 leading buffers along with 20 sampling flip-flops and their related buffers. Here, the AFN value is computed based on the flip-flop outcomes in 8 consecutive clock cycles. In these experiments, we didn't change the voltage and temperature, rather we show how the change of the system clock frequency can be caught by our sensor resulting in raising an alarm when the circuit operates out of spec.

Along with our sensor, we have implemented a round-based AES core (shown in blue in Fig. 17). The AES module has been implemented close to the sensor modules. The AES cipher used as a proof of concept and it can be replaced by any other target system. Here we want to show if the sensors can detect the clock manipulation in AES.

Sensors and AES all are fed with the same clock signal. We implemented 8 sensors as a proof of concept to investigate the impact of process variations in the sensors' outcome. The sensors were synthesized using Xilinx ISE and we performed place and route by ISE FPGA editor manually to make hard macros such that all sensors are similar in terms of placement and routing of the underlying components. Then the hard macros were instantiated throughout the FPGA yet close to the AES module. UART communication is used to communicate with the FPGA and the PC.

We used our AFN based sensor characterization to detect and report the fault attack realized by clock manipulation. Our attack model assumes that the attacker tries to force denial of service in AES by increasing the clock frequency beyond the highest limit, thus resulting in the circuit malfunction. Accordingly, we fed the AES with a random plaintext and ran the circuit at various clock frequencies. In each case, we collected the sensors' outcome (AFN value) along with the AES ciphertext to check if AES works properly at each of the considered frequencies or not.

The clock frequency was regulated in steps of 2 MHz from 70 to 100 MHz. We observed that AES will fail at clock frequencies beyond 86 MHz. In our experiments, this value was related to AFN=8.5 in our first sensor (Sensor-1) and very similar value in the other 7 sensors. Note that both AES and sensor operate under the same clock.

Figure 18 depicts the AFN variations in each of the 8 sensors when the circuitries operate at various frequencies. As shown the higher the frequency, the lower the AFN value. The parts shown in red in this figure relate to the conditions under which an alarm should be raised (AFN <8.5). In contrast, the grey areas denote the safe conditions. Note that we showed this outcome as a proof of concept confirming the usability of AFN characterization and the deployed sensor in detecting fault attacks.

Another observation that can be made from Fig. 18 is the similarity of the sensors' outcome confirming that the impact of process variation is negligible given that the sensor designs and placement and routing of the components embedded in each sensor are similar. In practice, to sense the voltage and temperature of the target circuitry the sensors should be placed close to the targeted circuits (AES in our case). Indeed in real industrial applications, multiple sensors are used and placed all over the chip to better sense the operating conditions of each part of the circuit. This is illustrated in Fig. 19 on the example of a security system-on-chip; the use-case is that representative of Common Criteria Protection Profile 0084 [40].

Indeed, the above experiment validates our digital sensor's ability to detect timing violations at the hardware level. Furthermore, we discovered that the sensor's

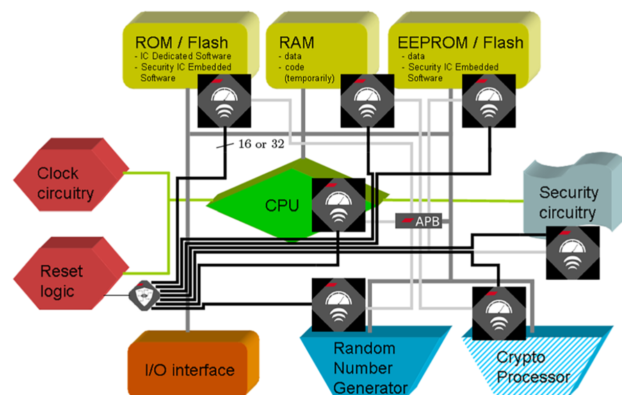


Fig. 19 Multiple digital sensors are instantiated, each in the vicinity of a sensitive resource; this allows for a large coverage of the chip various assets

characterization did not change dramatically in hardware as a result of process variation, with the overall range of AFN deviation being within 0.5 in the majority of cases. The key takeaway points from these observations is that the sensor can quickly detect timing violations caused by clock frequency manipulations in hardware, and the process variation has a minimal impact on its characterization.

9 Conclusion and Future Directions

In this paper we showed that digital sensors do age and the effect of such aging incurs a drift in the fault detection thresholds. Therefore, digital sensors require recalibration dynamically. Indeed, sensible deviations which can cause abnormal false positive and false negative shall be fixed in digital sensors. However, the prediction of such problems is difficult. In this respect, we proposed a very simple method to quantify the amount of such deviation: an idle sensor is instantiated, and the differential status between the functional and the idle sensor reveals the amount of discrepancy imposed by aging.

We proposed two calibration schemes. The first one is rather immediate: the active sensor thresholds are translated according to the observed difference between the idle and the active sensors. The second one relies on a more sophisticated approach, i.e., leveraging machine learning techniques. Both methods allow to successfully alleviate the effect of aging, hence to maintain low levels of false and missed alarm rates.

As a perspective, we intend to quantify the reliability of our both approaches, under the prism of safety. Indeed, for mission-critical applications, it is required to model accurately the FIT ("failure in time") rate as defined in the functional safety standard IEC 61508.

Acknowledgements This work has benefited from a funding via the bilateral project APRIORI (*Advanced PRivacy of IOT Devices through Robust Hardware Implementations*), from FR-DE cybersecurity 2020 call (*MESRI-BMBF*), managed by ANR from the French side. It has been also supported by the National Science Foundation CAREER Award (*NSF CNS-1943224*), and *NSF MRI* Award (1920079).

Data Availability All data generated or analyzed during this study are within the paper.

Declarations

Conflict of Interests The authors declare that they have no conflict of interest.

References

1. Anik MTH, Guilley S, Danger JL, Karimi N (2021) Detecting failures and attacks via digital sensors. IEEE Trans on Computer-Aided Design of Integrated Circuits and Systems (TCAD) 40(7):1315–1326
2. Ebrahimabadi M, Anik MTH, Danger JL, Guilley S, Karimi N (2020) Using digital sensors to leverage chips' security. In: Physical Assurance and Inspection of Electronics (PAINE), pp 1–6
3. Anik MTH, Guilley S, Danger JL, Karimi N (2020a) On the effect of aging on digital sensors. In: Proc. International Conference on VLSI Design (VLSID), pp 189–194
4. Anik MTH, Saini R, Danger J, Guilley S, Karimi N (2020b) Failure and attack detection by digital sensors. In: Proc. IEEE European Test Symposium (ETS), pp 1–2
5. Weste NH, Eshraghian K (1985) Principles of CMOS VLSI design: a systems perspective. NASA STI/Recon Technical Report A 85:47028
6. Brouchier J et al (2009) Temperature attacks. IEEE Secur Priv 7(2):79–82
7. Karimi N, Kanuparthi AK, Wang X, Sinanoglu O, Karri R (2015) MAGIC: malicious aging in circuits/cores. ACM Trans on Architecture and Code Optimization (TACO) 12(1):5:1–5:25
8. El Mrabet N, Page D, Vercauteren F. (2012) Fault attacks on pairing-based cryptography. pp 221–236, https://doi.org/10.1007/978-3-642-29656-7_13, Chapter 13 of [18]
9. Alam MM, Tajik S, Ganji F, Tehranipoor M, Forte D (2019) RAM-Jam: Remote temperature and voltage fault attack on FPGAs using memory collisions. In: Proc. Fault Diagnosis and Tolerance in Cryptography (FDTC), pp 48–55
10. Krautter J, Gnad DR, Tahoori MB (2018) FPGAhammer: Remote voltage fault attacks on shared FPGAs, suitable for DFA on AES. Trans on Cryptographic Hardware and Embedded Systems (CHES) pp 44–68
11. Murdock K, Oswald D, Garcia FD, Van Bulck J, Gruss D, Piessens F (2020) Plundervolt: Software-based fault injection attacks against Intel SGX. In: Proceedings of the 41st IEEE Symposium on Security and Privacy (S&P'20), pp 1466–1482
12. Tang A, Sethumadhavan S, Stolfo SJ (2017) CLKSCREW: exposing the perils of security-oblivious energy management. In: Kirda E, Ristenpart T (eds) 26th USENIX security symposium, USENIX security 2017, Vancouver, BC, Canada, August 16-18, 2017, USENIX Association, pp 1057–1074, https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/tang
13. Agoyan M, Dutertre JM, Naccache D, Robisson B,Tria A (2010) When clocks fail: on critical paths and clockfaults. In: CARDIS, lecture notes in computer science,vol. 6035, pp 182–193, Springer, Passau, Germany
14. Selmane N, Guilley S, Danger JL (2008) Practical setup time violation attacks on AES. In: Proc. Seventh european dependable computing conference, pp 91–96
15. Qiu P, Wang D, Lyu Y, Qu G (2019) Voltjockey: Breaching trustzone by software-controlled voltage manipulation over multi-core frequencies. In: Proc. Conf. on Computer and Communications Security (CCS), pp 195–209
16. Kenjar Z, Frassetto T, Gens D, Franz M, Sadeghi AR (2020) VOLTpwn: Attacking x86 processor integrity from software. In: Proc. USENIX Security Symp., pp 1445–1461
17. Chen Z, Vasilakis G, Murdock K, Dean E, Oswald D, Garcia FD (2021) VoltPillager: Hardware-based fault injection attacks against Intel SGX Enclaves using the SVID voltage scaling interface. In: Proc. USENIX, pp 699–716
18. Shoukry Y, Martin PD, Tabuada P, Srivastava MB (2013) Non-invasive spoofing attacks for anti-lock braking systems. In: Proc. Conf. on Cryptographic Hardware and Embedded Systems (CHES), pp 55–72
19. Korak T, Hoefler M (2014) On the effects of clock and power supply tampering on two microcontroller platforms. In: Proc. Workshop on Fault Diagnosis and Tolerance in Cryptography, (FDTC), pp 8–17

20. Korak T, Hutter M, Ege B, Batina L (2014) Clock glitch attacks in the presence of heating. In: Proc. Workshop on Fault Diagnosis and Tolerance in Cryptography, pp 104–114
21. Joye M, Tunstall M (eds) (2012) Fault analysis in cryptography. Springer
22. Amrutur B, Mehta N, Dwivedi S, Gupte A (2011) Adaptative techniques to reduce power in digital circuits. J Low Power Electron Appl 1(2):261–276
23. Guilley S, Newell R, Porteboeuf T (2014) Reliability analysis of digital sensors against perturbations of FPGAs. In: Proc. 12th CryptArchi Workshop
24. Selmane N, Bhasin S, Guilley S, Danger JL (2011) Security evaluation of application-specific integrated circuits and field programmable gate arrays against setup time violation attacks. IET Inf Secur 5(4):181–190
25. Selmane N, Guilley S, Danger JL (2008) Setup time violation attacks on aes. In: Proc. European Dependable Computing Conf. (EDCC), pp 91–96
26. NIST FIPS (Federal Information Processing Standards) publication 140-3 (2019) Security requirements for cryptographic modules. https://csrc.nist.gov/publications/detail/fips/140/3/final
27. Martin AJ, Nyström M (2006) Asynchronous techniques for system-on-chip design. Proc IEEE 94(6):1089–1120
28. Shahrjerdi D, Rajendran J, Garg S, Koushanfar F, Karri R (2014) Shielding and securing integrated circuits with sensors. In: Proc. ICCAD, pp 170–174
29. De Marcellis A, Ferri G (2011) Analog circuits and systems for voltage-mode and current-mode sensor interfacing applications. Springer
30. van der Horn G, Huijsing JL (2012) Integrated smart sensors: design and calibration, vol 419. Springer
31. Guilley S, Danger JL (2012) Global faults on cryptographic circuits. Chapter 17 of [18], pp 295–311
32. El-Baze D, Rigaud J, Maurine P (2016a) An embedded digital sensor against EM and BB fault injection. In: Workshop on Fault Diagnosis and Tolerance in Cryptography, (FDTC), pp 78–86
33. El-Baze D, Rigaud J, Maurine P (2016b) A fully-digital EM pulse detector. In: Design, Automation & Test in Europe (DATE), pp 439–444
34. Oboril F, Tahoori MB (2012) Extratime: Modeling and analysis of wearout due to transistor aging at microarchitecture-level. In: Proc. International Conference on Dependable Systems and Networks (DSN), pp 1–12
35. Karimi N, Danger JL, Guilley S (2018) Impact of aging on the reliability of delay PUFs. J Electron Test Theory Appl (JETTA) 34(5):571–586
36. Alam MA, Kufluoglua H, Varghese D, Mahapatra S (2007) A comprehensive model for PMOS NBTI degradation: Recent progress. Microelectron Reliab 47(6):853–862
37. Khan S, Haron NZ, Hamdioui S, Catthoor F (2011) NBTI monitoring and design for reliability in nanoscale circuits. In: Proc. Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS), pp 68–76
38. Guilley S, Facon A, Bruneau N (2021) Quantitative digitalsensor 2021, Patent US16/954,507 (applicationUS20210004461A1, pending)
39. Nangate 45nm open cell library. "http://www.nangate.com" (last accessed 9 Dec.,2020)
40. Bundesamt für Sicherheit in der Informationstechnik (2014) SI-CC-PP-0084-2014: Security IC platform protection profile with augmentation packages. Version 1.0. https://www.commoncriteriaportal.org/files/ppfiles/pp0084a_pdf.pdf

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Md Toufiq Hasan Anik received the B.Sc. degree in Electrical and Electronics Engineering from BRAC University, Bangladesh in 2016. He is currently a Ph.D. student in Computer Engineering at University of Maryland Baltimore County (UMBC). He worked at Intel as a Computer Architecture Graduate Intern during summer 2021 and as a Security Researcher Intern during summer 2020. His research interest includes hardware security and in particular, power analysis attacks and countermeasures, as well as sensor-assisted secure and reliable design. He conducts research in the SECure, REliable and Trusted Systems (SECRETS) research lab at UMBC.

Mohammad Ebrahimabadi received the B.Sc. degree in Electrical Engineering from Zanjan University, Zanjan, Iran, in 2008, and the M.Sc. degree in Electrical Engineering from the Sharif University of Technology, Tehran, Iran, in 2011. He is working towards the Ph.D. degree in the Department of Computer Science and Engineering at the University of Maryland Baltimore County, MD, USA since 2019. He is a member of the SECure, REliable and Trusted Systems (SECRETS) research lab. His current research focus is on developing PUF-based Authentication and Secure Communication Protocols in IoT networks, as well as sensor-assisted secure and reliable design.

Jean-Luc Danger is full Professor at TELECOM Paris. He is the head of the digital electronic system research team involved in Research in security/safety of embedded systems, configurable architectures, and implementation of complex algorithms in ASICs or FPGAs. He authored more than 250+ scientific publications and patents in architectures of embedded systems and security, and is the co-founder & scientific advisor of the Secure-IC company. He received his engineering degree in Electrical Engineering from École Supérieure d'Électricité in 1981. After 12 years in industrial laboratories (PHILIPS, NOKIA), he joined TELECOM ParisTech in 1993 where he became full professor in 2002. He is a co-founder of Secure-IC. His personal research interests are trusted computing, cyber-security, random number generation, and protected implementations in novel technologies.

Sylvain Guilley is General Manager and Chief Technology Officer at Secure-IC, a company offering security for embedded systems. Secure-IC's flagship technology is the multi-certified Securyzr® integrated Secure Element (iSE). Within Secure-IC, he is also director of "Threat Analysis" and "Think Ahead" business lines, which develop respectively security evaluation tools and advanced research. Sylvain is also professor at TELECOM-Paris, associate research at École Normale Supérieure (ENS), and adjunct professor at the Chinese Academy of Sciences (CAS). His research interests are trusted computing, cyber-physical security, secure prototyping in FPGA and ASIC, and formal/mathematical methods. Since 2012, he organizes the PROOFS workshop, which brings together researchers whose objective is to increase the trust in the security of embedded systems. He is also lead editor of international standards, such as ISO/IEC 20897 (Physically Unclonable Functions), ISO/IEC 20085 (Calibration of non-invasive testing tools), and ISO/IEC 24485 (White Box Cryptography). He is "High Level Principles for Design/Architecture" team leader for the drafting of Singapore TR68 standard on Cyber-Security of Autonomous Vehicles. Sylvain is associate editor of the Springer Journal of Cryptography Engineering (JCEN). He has coauthored 250+ research papers and filed 40+ patents. He is member of the CMUF, the IACR and senior member of the IEEE and of the CryptArchi club. He is an alumni of Ecole Polytechnique and TELECOM-ParisTech.

Naghmeh Karimi received the B.Sc., M.Sc., and Ph.D. degrees in Computer Engineering from the University of Tehran, Iran in 1997, 2002, and 2010, respectively. She was a visiting researcher at Yale University, USA between 2007 and 2009, and a post-doctoral researcher at Duke University, USA during 2011-2012. She has been a visiting assistant professor at New York University and Rutgers University between 2012

and 2016. She joined University of Maryland Baltimore County as an assistant professor in 2017 where she leads the SECure, REliable and Trusted Systems (SECRETS) research lab. She has published three book chapters and authored/co-authored more than 70 papers in referred conference proceedings and journal manuscripts. She serves as an Associate Editor of the Springer Journal of Electronic Testing: Theory and Applications (JETTA). She is also the corresponding guest editor of the Journal on Emerging and Selected Topics in Circuits and Systems (JETCAS); special issue in Hardware Security in Emerging Technologies. Her current research interests include hardware security, VLSI testing, design-for-trust, design-for-testability, and design-for-reliability.