Connecting the Physical and Application Level Towards Grasping Aging Effects

Hussam Amrouch[1], Javier Martin-Martinez[2], Victor M. van Santen[1],
Miquel Moras[2], Rosana Rodriguez[2], Montserrat Nafria[2], Jörg Henkel[1]
[1]Karlsruhe Institute of Technology, Chair for Embedded Systems (CES), Karlsruhe, Germany
[2]Universitat Autonoma de Barcelona (UAB), Department of Electronic Engineering, Barcelona, Spain
{amrouch, victor.santen, henkel}@kit.edu, {javier.martin.martinez, miquel.moras, rosana.rodriguez, montse.nafria}@uab.cat

***Abstract*—Technology scaling noticeably increases the susceptibility of transistors to varied degradations induced by aging phenomena like *Bias Temperature Instability* (BTI) and *Time-Dependent-Dielectric Breakdown* (TDDB). Therefore, estimating the reliability of an entire computational system necessitates investigating how such phenomena will ultimately lead to failures – considering that aging starts from the physical level and ends up at the application level, where workloads (i.e. software programs) run. The key challenge is that an accurate estimation imposes analyzing the impact of aging on each individual transistor within a sophisticated on-chip system using complex physics-based models. The latter requires both a careful experimental model parameter derivation for calibration and precise information regarding the actual temperature voltage-stress waveforms that may be applied to the transistors during lifetime. These waveforms are directly *driven* by the running workloads creating the inevitable necessity to connect the physical and application level. As a matter of fact, this challenge is exacerbated in the nano era, due to the *typical* workloads (i.e. multiple applications running in parallel along with an operating system) that may run on top of a tremendous number of transistors. This paper investigates this challenge to provide designers with an abstracted, yet accurate reliability estimation that takes into account the interrelations between the physical and application level towards grasping how aging actually degrades on-chip system reliability.**

I. Introduction

Transistors in deep nano scale became increasingly susceptible to aging effects due to higher electric fields and current densities that lead to increasing defect activation rates within the gate dielectric of transistors. These defects degrade the transistor characteristics (e.g., V_{th}) and ultimately jeopardize the reliability of the entire on-chip system. Defect activation strongly depends on the applied temperature and voltage-stress waveforms to the transistor (i.e. higher temperature and/or voltage accelerate aging). Importantly, these waveforms are directly *driven* by running applications as their activities (access patterns to the micro-architectural components of the on-chip system) play a substantial role in defining the transistor switching rates. For instance, high-activity applications lead to a higher dynamic power consumption that, in turn, results in a higher temperature. Therefore, grasping aging effects necessitates accurately estimating the actual temperature/voltage-stress waveforms which cannot be independent from investigating the running applications. Otherwise, designers need to assume worst-case scenarios to circumvent the lack of these waveforms leading to exaggerating the strength of aging-induced degradations and thus overestimating the overall impact of aging – especially, compared to the consideration of actual waveforms extracted from *typical* scenarios (i.e. workloads similar to the end-user software of on-chip systems).

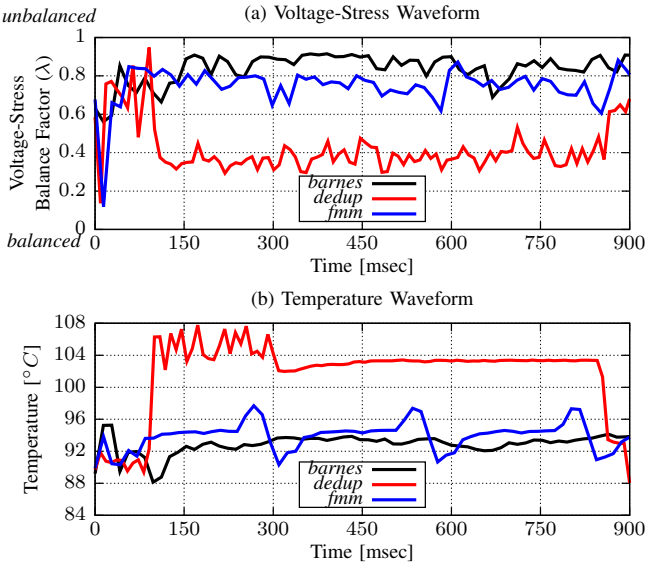

Fig. 1: (a) Highlights the differences between the voltage-stress waveforms from three applications. (b) Shows the corresponding temperatures waveforms and how running applications may induce a high (e.g., *fmm*) or low (e.g., *barnes*, *dedup*) temporal variation within the waveform. Note that each data point (within the voltage-stress waveform) represents λ for an interval of 0.1 msec

Our key contributions within this paper are:
We investigate how applications *drive* aging through the extraction of the induced temperature/voltage-stress waveforms and then passing them to a set of physical aging models, that have been accurately calibrated to reflect the impact of different waveforms on transistor. Afterwards, the induced degradations are *abstracted*, from the physical up to the application level, where they can be *grasped* (i.e. interpreted as system failures). This extends state-of-the-art where the impact of aging is primarily analyzed from the physical up to solely the circuit level [2], [3]. Additionally, state-of-the-art often considers worst-case operating conditions (e.g., V_{DD} $\geq$1.5V, T=125°C, constant high voltage-stress waveform), when investigating aging [4], [5]. Whilst, we focus on the typical operating conditions, that may be actually induced

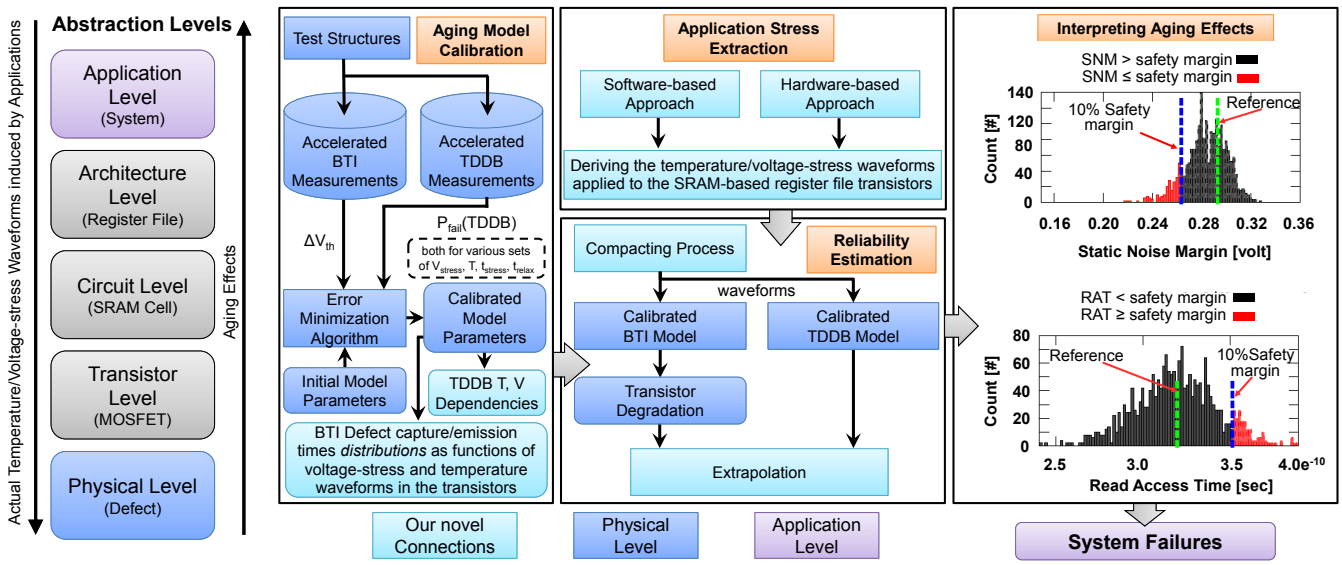

Fig. 2: Overview of our proposed connection to grasp aging effects. On the left, the connected abstraction levels are shown. In the middle the aging model calibration and stress waveforms extraction are presented. The right side illustrates the interpretation of aging-induced degradations in SRAMs (e.g., SNM and RAT) as failures, based on the safety margin concept [1] (further details in Section II-E)

by running applications, towards obtaining accurate failure analysis, representing when/where aging fails the system. This is due to the fact that different applications can result in different temperature/voltage-stress waveforms (see Fig 1) and thus taking them into account is inevitable to grasp aging effects.

To exemplify our approach, we select SRAM cells as they are vulnerable to a variety of hazards causing failures due to both data corruption and timing violations. This is more comprehensive than other examples that have often been used in state-of-the-art (e.g., ring oscillators [3]) because the latter solely suffers from a single kind of failures (i.e. timing violations). SRAMs can be employed to implement different microprocessor components such as register files (e.g., Intel WSM core [6]). In fact, register files are recognized as one of the reliability-critical processor components [7], due to their high utilization, and they are particularly susceptible to aging because of elevated temperatures [8] and/or unbalanced [9] voltage-stress waveforms.

Therefore, the register file is one of the first-class candidates to study the connection between the physical and application level in order to grasp how aging fails on-chip systems.

Prerequisites: To briefly discuss a standard 6T SRAM cell and its characteristics: it contains two cross-coupled inverters and thus while a logic value is stored, transistors diagonal to each other (i.e. pMOS of left-side inverter along with nMOS of right-side inverter) are constantly under *voltage-stress*, while the other transistors, forming the second diagonal, are in *relaxation*. The voltage-stress waveform of an SRAM describes its *duty cycle*, i.e. the stress/relaxation ratio for its transistors. To abstract from the duty cycle, we define the Voltage-Stress Balance Factor (λ) to represent how the *voltage-stress* is being balanced among the SRAM transistors during a specific interval of time. In practice, $\lambda \in [0, 1]$ and $\lambda = 0$ indicates that the *voltage-stress* has been equally distributed in that interval and thus aging has evenly degraded the SRAM transistors. On the other hand, $\lambda = 1$ indicates that solely one transistors diagonal has been stressed (i.e. the SRAM continuously stores the same logic value). $\lambda_{overall}$, in turn, indicates the voltage-stress balance factor for the *entire* application's execution.

One of the key reliability metrics for SRAMs is the Static Noise Margin (SNM) which represents its resiliency against noise and thus its resiliency against failures due to data corruption. In addition, the Read Access Time[1] (RAT) represents the timing behavior of SRAM and thus implicitly its resiliency against failures due to timing violations. Therefore, grasping the impact of aging on SRAM-based components necessitates analyzing how aging degrades both SNM and RAT.

II. Our Proposed Approach of Connecting the Physical and Application Level

Fig 2 presents an overview of how the application-induced temperature/voltage-stress waveforms are extracted and then passed to the employed physical aging models. Afterwards, the aging-induced degradations stimulated by these waveforms are *abstracted*, from the physical up to the application level, where they can be *interpreted* as system failures. In the following, we demonstrate step by step our proposed approaches to connect the physical and application level towards grasping aging effects.

A. Temperature/Voltage-Stress Waveforms Extraction

The activities (e.g., read/write accesses to the register file) at the *application level* need to be monitored during the execution of the applications to extract the actual temperature/voltage-stress waveforms that are applied to the transistors of each

[1]Write Access Time improves with aging and hence is not studied.

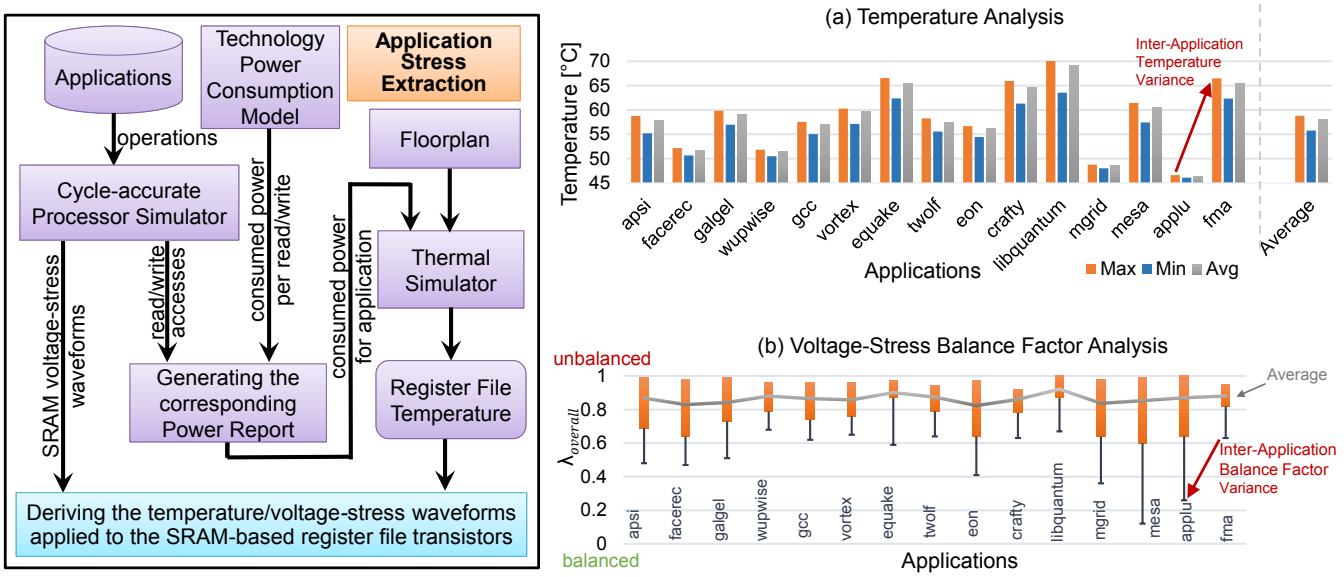

Fig. 3: Our software-based approach, where the extraction of voltage-stress waveforms is performed through an instruction set simulator that also monitors the applications activities. Based on these activities, the corresponding on-chip power densities are estimated and then used by a thermal simulator to estimate the temperature. (a) Steady-state temperature analysis for various applications from the SPEC benchmarks suite individually running on bare metal. (b) presents the $\lambda_{overall}$ analysis of the corresponding applications showing the inter-application and intra-application (error-bars) variation of the entire application

6T SRAM within the register file. In embedded on-chip systems, applications run either without an operating system (i.e. bare metal) or along with an operating system (e.g., Linux) that manages the running of multiple applications in parallel through its scheduler. Therefore, in this work, we developed *software* and *hardware*-based approaches to connect the physical and application level according to the requirements of the studied on-chip system.

Software-based Approach: In this approach, we employ simulation tools to extract the induced temperature/voltage-stress waveforms. First, an instruction set simulator of the processor architecture (e.g., sim-alpha, gem5 [10]) monitors the activities of the running applications (read/write access traces) as well as it calculates the induced λ in each single SRAM within the register file. Then, a technology power consumption simulator (e.g., McPAT [11]) provides us, according to the desired technology node (e.g., 45nm, 22nm), with the estimated power consumption per each read/write access to the register file. Afterwards, the power consumption waveforms are calculated and then passed to a thermal simulation (e.g., HotSpot [12]) to calculate the corresponding temperature waveforms. In our experiments, we employed diverse applications – exhibiting different behaviors – from the PARSEC and SPEC benchmark suites [13], [14], which both are widely used for the Alpha superscalar architecture [2] employed in the evaluation of this work. Fig 3 illustrates the implementation along with the obtained analysis for the case of applications from SPEC executed on bare metal. Such an approach, while being general (i.e. applicable to any architecture kind), unfortunately comes with a high computational time that is directly related to the complexity of the simulated workload. For instance in a reasonable time, this approach can perform the required extraction for applications, with small input data, running on bare metal or for a region of interest (i.e. not the full execution) of an individual application running on top of an operating system. In other words, the software-based approach is only suitable to analyze a *superficial* (i.e. non-typical) workload as *typical* workloads contain an order of magnitude more operations and thus simulating them may not be feasible due to computational intensity.

Hardware-based Approach: To tackle the challenge of extracting the temperature/voltage-stress waveforms under *typical* workloads (i.e. monitoring the entire execution of multiple parallel applications running on top of an operating system), we developed a novel hardware-based approach[3]. First, we implement the available Register-Transfer Level (RTL) hardware design of the LEON3 on-chip system [15][4] in a Xilinx Virtex-5 FPGA platform. Then, we implement our own hardware component to monitor the activities – within the register file bits – induced by *typical* workloads. *This hardware monitor forms, in practice, our key novel contribution within this approach.* The challenge behind implementing this monitor is that several aspects need to carefully be taken into account. First, the processor critical path should not be negatively influenced. Otherwise, the on-chip system may crash. Secondly, the limited available area within the FPGA imposes an additional constraint restricting the monitor's implementation, i.e. the FPGA must jointly implement both the entire on-chip system along with our monitor component.

[2]Alpha has a register file of 80 registers with a bit-width of 64 bits.

[3]Only applicable for architectures with an available RTL design.

[4]LEON3 features a register file of 136 registers with bit-width of 32 bits.

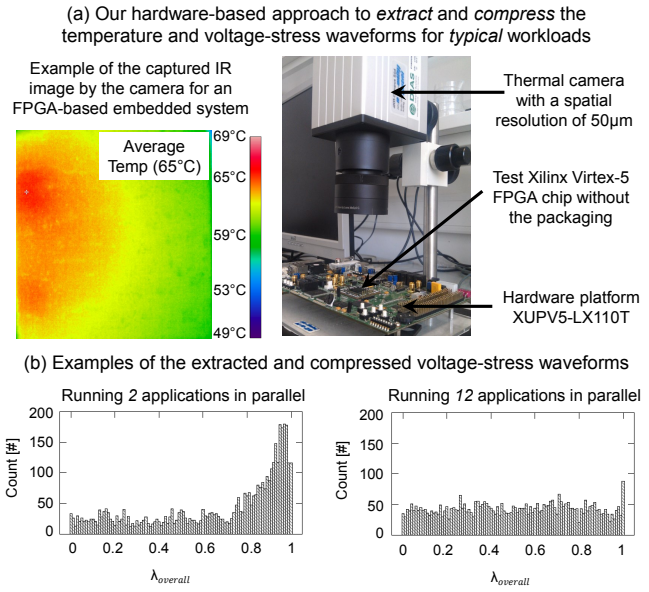

Fig. 4: (a) Our hardware-based approach for *typical* workloads, where a hardware monitor is implemented in a reconfigurable fabric (i.e. FPGA) along with the LEON3 processor [15] that runs Linux 2.6. The monitor extracts the voltage-stress waveforms induced by the workload and a thermal camera-based measurement setup [16] is employed to capture the chip's IR images and extract the temperature waveforms. (b) The $\lambda_{overall}$ distribution of the register file SRAMs extracted from real-time executions of multiple applications in parallel, highlighting the ability of our hardware platform to run *typical* workloads (e.g., 12 parallel applications)

It is also noteworthy that employing a hardware platform to run workloads provides a significant speed up of 35x compared to the employment of an instruction set simulator [17]. This makes analyzing *typical* workloads feasible. In these experiments, we utilized a set of embedded system applications from the Mibench benchmark suite [18] to run in parallel on top of the Linux 2.6 operating system as Fig 4(b) presents. Finally, the temperature waveforms are extracted by a thermal camera that accurately captures the IR emissions of the FPGA chip [16].

B. Waveforms Compression

As a matter of fact, the analyzed processor's component (e.g., register file) contains a significant number of transistors and, additionally, the processor executes billions of instructions per second. Therefore, the extraction process will provide a significant number[5] of waveforms each containing a massive number of data points. This makes employing detailed physical aging modeling unfeasible due to the resulting computational complexity. Thus, we developed a *compacting* process to compress these waveforms into equivalent ones with *significantly* truncated data points, i.e. lowering the unnecessary high temporal resolution of the waveform, while retaining the key information for the aging models. For voltage-stress waveforms, equivalent waveforms means waveforms with indistinguishable λ, as aging depends on the stress/relaxation ratio, while the dependence on the frequency of stress/relaxation cycles can be abstracted away. The physical principles behind our *compacting* process are explanation in section II-C. For temperature waveforms, we can truncate the amount of data points without jeopardizing the accuracy of our reliability estimation as temperature shifts are slow due to intrinsic thermal capacitance inhibiting fast changes.

Importantly, our process provides an enormous simulation speedup ($> 10^6$x), which is imperative to deal with aforementioned challenge of complexity. Within the software-based approach, the *compacting* process has been developed as a software algorithm, while our hardware monitor within the hardware-based approach performs on-the-fly compression of the obtained voltage-stress waveforms before transmitting them out of the FPGA platform.

C. Physics-based BTI Model Calibration

The compressed temperature/voltage-stress waveforms will serve as inputs for a physics-based aging model to calculate the induced degradation in the threshold voltage of the analyzed transistor (ΔV_{th}). To this end, we employed a recent BTI model that is based on the assumption that BTI effects are related to charge capturing/emission of defects within the transistor leading to an increase/decrease of the V_{th} [4], [5]. While, V_{th} increases under high voltage conditions (capturing charges in the stress phase), it decreases when voltage is lowered (emission of charges in the relaxation phase). Whereas, high temperature conditions accelerate both capturing/emission of charges.

The processes of capturing/emitting charges depend on the properties of the acting defects, which require three parameters to be individually described: (1) The defect charge capture and (2) emission times (τ_c and τ_e, respectively), which are voltage and temperature dependent. (3) The ΔV_{th} associated to the capturing/emission of charges in/from that defect ($\eta = \Delta V_{th}(\text{defect})$). To calculate the ΔV_{th} of the transistor, all the defects within the transistor and their occupation probability

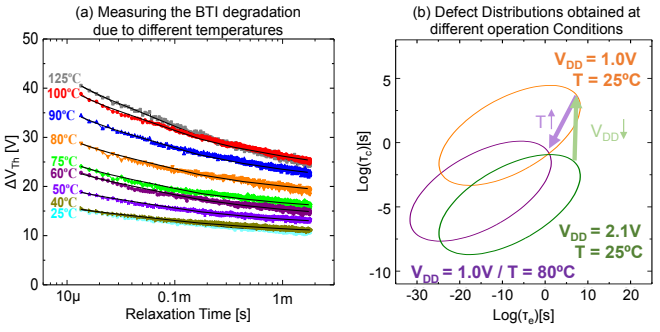

Fig. 5: (a) Measured $\Delta V_{th}(t_r)$ in pMOS transistors at $T \in [25°C, 125°C]$ after 9ms stress at $V_{stress} = -2.1V$. Lines are model calibrated to fit the measurement $D(\tau_e, \tau_c)$ shown in (b).
(b) Capture/Emission time defect distributions $D(\tau_e, \tau_c)$ for various operating conditions, highlighting shorter capture times at higher V_{DD} and shorter capture/emission times at higher T $D(2.1V/25°C)$ is measured [19], while the others are examples of extrapolated D based upon a set of measurements at various conditions (V, T)

[5] >25000 for LEON3, >30000 for Alpha assuming 6T SRAM

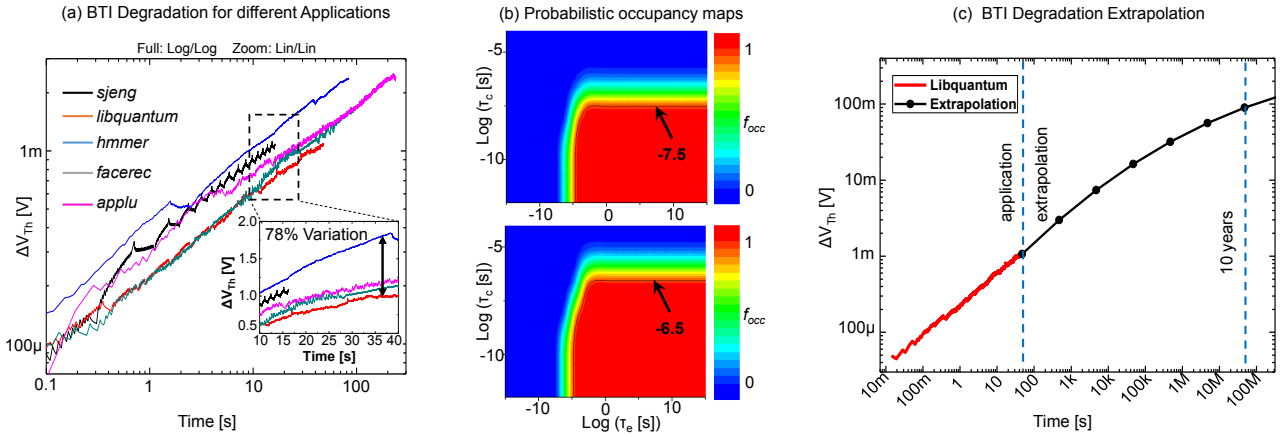

Fig. 6: (a) BTI-induced ΔV_{th} for different applications from the SPEC benchmark. Note the high variation in the lin/lin magnification, which is hidden in the overall log/log plot. (b) f_{occ} calculated for 100 (top) and 1000 periods (bottom) of a 100 kHz pulsed waveform, black line corresponds to $f_{occ} = 0.95$. A 10x increase in the number of periods results in a decade shift in f_{occ}. (c) Extrapolation of the BTI-induced ΔV_{th} for the periodic execution of the the *libquantum* application over the typical lifetime of 10 years

(i.e. probability of a captured charge occupying a defect) must be considered. Therefore, ΔV_{th} can be calculated – for any stress (t_s) and relaxation (t_r) times – as follows:

$$\Delta V_{th}(t_s, t_r) = N \cdot \overline{\eta} \int_0^\infty \int_0^\infty D(\tau_e, \tau_c) \cdot f_{occ}(\tau_c, \tau_e; t)\, \mathrm{d}\tau_e \mathrm{d}\tau_c \quad (1)$$

$$\text{with} \tau_c = \tau_c(V,T) \text{ and } \tau_e = \tau_e(V,T)$$

Where, N is the number of defects in the transistor, $\overline{\eta}$ is the mean value of the η's of all the defects, $D(\tau_e, \tau_c)$ is the distribution of defects in the $\tau_c - \tau_e$ space, and $f_{occ}(\tau_c, \tau_e; t)$ is the probability that a defect with parameters τ_c and τ_e is occupied at a given time t and voltage/temperature conditions. f_{occ} is calculated using the procedure explained in [20]. A log-normal bi-variant distribution is assumed for $D(\tau_e, \tau_c)$ [4], [19], which depends on the fabrication technology, voltage V and temperature T. Then, to correctly estimate the ΔV_{th} induced by BTI, it is crucial to obtain the suitable $D(\tau_e, \tau_c)$, for the considered technology and transistor operation conditions, i.e the model must be calibrated. For a certain technology, the parameters (i.e. τ_c / τ_e) defining the bi-variant defect distribution D, are obtained from measurements under accelerated conditions (i.e. elevated temperatures and high $V = V_{stress}$) on special technology test structures. The parameters are chosen to fit the V_{th} evolution of the transistors during the measurements performed within feasible short stress times due to the accelerated conditions. Then the aforementioned process must be repeated for various T and V_{stress} combinations to calibrate the temperature and voltage dependency of the model. This is essential to connect the physical and application level as running applications exhibit a wide range of temperatures/voltage-stress waveforms (see Fig 3(a,b)). To exemplify the procedure, BTI-induced ΔV_{th} observed at different voltages and temperatures ($V_{stress} \in [-1.8V, -2.4V]$ and $T \in [25°C, 125°C]$) were measured in pMOS transistors and fitted to equation 1.

Fig 5(a) shows an example of the ΔV_{th} shifts obtained during relaxation after a 9ms stress at $V_{stress} = -2.1V$, for $T \in [25°C, 125°C]$. The continuous lines correspond to the fitting of the data to equation 1. With fitting, the parameters of $D(\tau_e, \tau_c)$ which minimize the error between experimental data and eq 1 are obtained, as a function of (V_{stress}, T). Fig 5(b) shows the obtained defect distribution D for the data in Fig. 5(a) at $25°C$ (green ellipse). These distributions D can be extrapolated to actual operation conditions (i.e. lower voltages (orange ellipse) and higher temperatures (purple ellipse)), by considering the observed experimental dependence of the defect distribution on voltage and temperature as discussed in [19].

The BTI model takes stress and relaxation phases into account to estimate the temporal evolution of ΔV_{th} based on the corresponding extracted voltage-stress waveforms for that transistor. Fig 7(a) shows several examples of the estimated V_{th} evolutions when the transistor is subjected to various

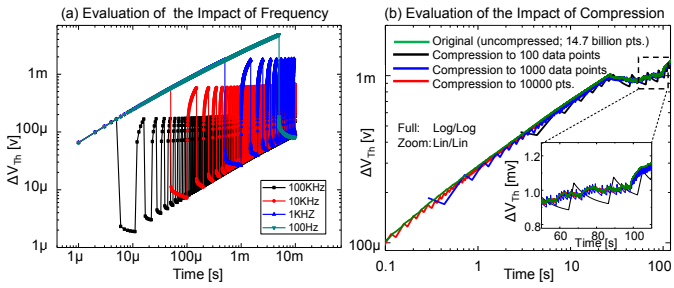

Fig. 7: (a) Frequency independence of ΔV_{th} on the lower edge of the data points, where $\lambda(f,t)$ is equal for all points. Other data points (where $\lambda(f_1,t) \neq \lambda(f_2,t)$) are frequency dependent. (b) Comparison of the induced V_{th} degradation due to various compression ratios (e.g., compressing the original waveform to 100, 1000 or 10000 data points) against the original (i.e. uncompressed) waveform. As shown, compression introduces a negligible error. As the new waveforms still lead to a very similar V_{th} degradation to the one induced by the original waveform

waveforms with different frequencies. Note that, though aging depends on the stress frequency [4], the ΔV_{th} that will be measured after a short relaxation time is frequency independent, as experimentally observed [21]. *In practice, this property forms the fundamental scientific core of our compacting process presented in Section II-B.* Fig 7(b) illustrates how such equivalent waveforms, obtained from our developed *compacting* process, result in similar ΔV_{th} to the original (i.e. uncompressed) waveform with negligible errors –*as long as the compression guarantees* $\lambda(compression) = \lambda(original)$– as the magnification highlights.

D. Extrapolation Towards Standard On-chip Lifetimes

Fig 6(b) shows the ΔV_{th} induced by different applications and demonstrates that distinct applications differently degrade V_{th} ($> 70\%$). It is also noteworthy that such analysis shown in Figs (6(b), 5(a)) are only feasible for short time windows, even though the lifetime of on-chip systems is typically several years (e.g., 10 years). Therefore, extrapolating the obtained results is necessary to grasp how aging effects ultimately influence on-chip systems. As a matter of fact, under periodic conditions, $f_{occ}(t)$ logarithmically shifts with time, as shown in Fig 6(a). Since $f_{occ}(t)$ is the only term in Eq. 1 that depends on time t, once $f_{occ}(t)$ has been calculated for a large number of periods, the ΔV_{th} can be obtained. Periodic stress conditions are valid, since periodic triggering of the same applications is typical in embedded on-chip systems which are our target. This enables us to extrapolate properly from minutes to the typical on-chip system lifetime, as presented in Fig 6(c).

E. Interpretation of Aging-induced Degradation

After estimating the BTI-induced V_{th} degradation in all transistors within the register file SRAMs as earlier described, we employ an *abstraction* process in order to *interpret* how these degradations ultimately induce failures (i.e. calculating P_{fail}) due to data corruption and timing violations, based on our recent work [1]. In that work, the aging effects from physical up to application level (see Fig 2(left)) has been *abstracted* and *interpreted* towards a probabilistic fault analysis based on the safety margin concept. This concept assumes that the SRAM circuit has been over-designed by a certain degree to provide a tolerance regarding the varied degradations that may be induced within the on-chip system during its lifetime. Thus, as soon as the degradation exceeds the predetermined safety margin, failures start to occur as the SRAM starts to operate worse than its specification. For instance, a typical safety margin of $S = 10\%$ indicates that the on-chip system will exhibit failures if the aging degrades the SNM and RAT of SRAMs by more 10%.

$$P_{fail}(SNM) = |i \in SRAM : SNM(aged, i) < SNM(fresh, i) * (1 - S)|$$
$$P_{fail}(RAT) = |i \in SRAM : RAT(aged, i) > RAT(fresh, i) * (1 + S)|$$

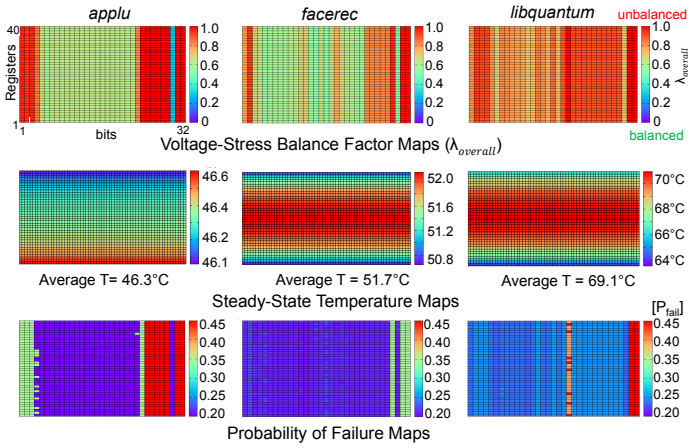

Fig. 8: Three applications from the SPEC suite executed on bare metal through our software-based approach. The top row shows the spatial distribution of $\lambda_{overall}$ for each SRAM cell. The middle row highlights the spatial distribution of temperature T across the register file. The bottom row illustrates the corresponding spatial distribution of P_{fail}

In addition to estimating the P_{fail} due to BTI-induced degradations, an empirical TDDB model [22] has been also employed to calculate the probability of failure due to TDDB. In such a case, the safety margin concept is not used as the TDDB model directly calculates $P_{fail}(TDDB)$ for every transistor within the SRAM assuming that all of them are critical for the SRAM to properly perform its operations.

$$P_{fail}(TDDB) = \sum_{i=0}^{\#transistors} P_{fail}(TDDB, i)$$

It is noteworthy that due to the overlap between the failure causes (as [1] established):

$$P_{fail}(System) \neq P_{fail}(SNM) + P_{fail}(RAT) + P_{fail}(TDDB)$$

For instance, a degraded SRAM may simultaneously fail due to SNM and RAT, yet still only counts as 1 erroneous cell. Finally, the total probability of failure due to both BTI and TDDB in an SRAM can be calculated by combining the failures due to them along with considering their interrelation. This, in turn, provides the failure probability map of the entire register file. Such a map demonstrates the ultimate impact of aging due to the running workload.

III. Evaluation

Our proposed idea of connecting the physical and application level (see Fig 2(left)) necessitates a twofold evaluation. First, we connect the application down to physical level by extracting the impact of applications on aging. Then, we connect the physical up to application level by interpreting how aging effects induce failures.

Application to physical level: First, we employed our software and hardware-based approaches explained in Sections II-A to extract the voltage-stress waveforms induced

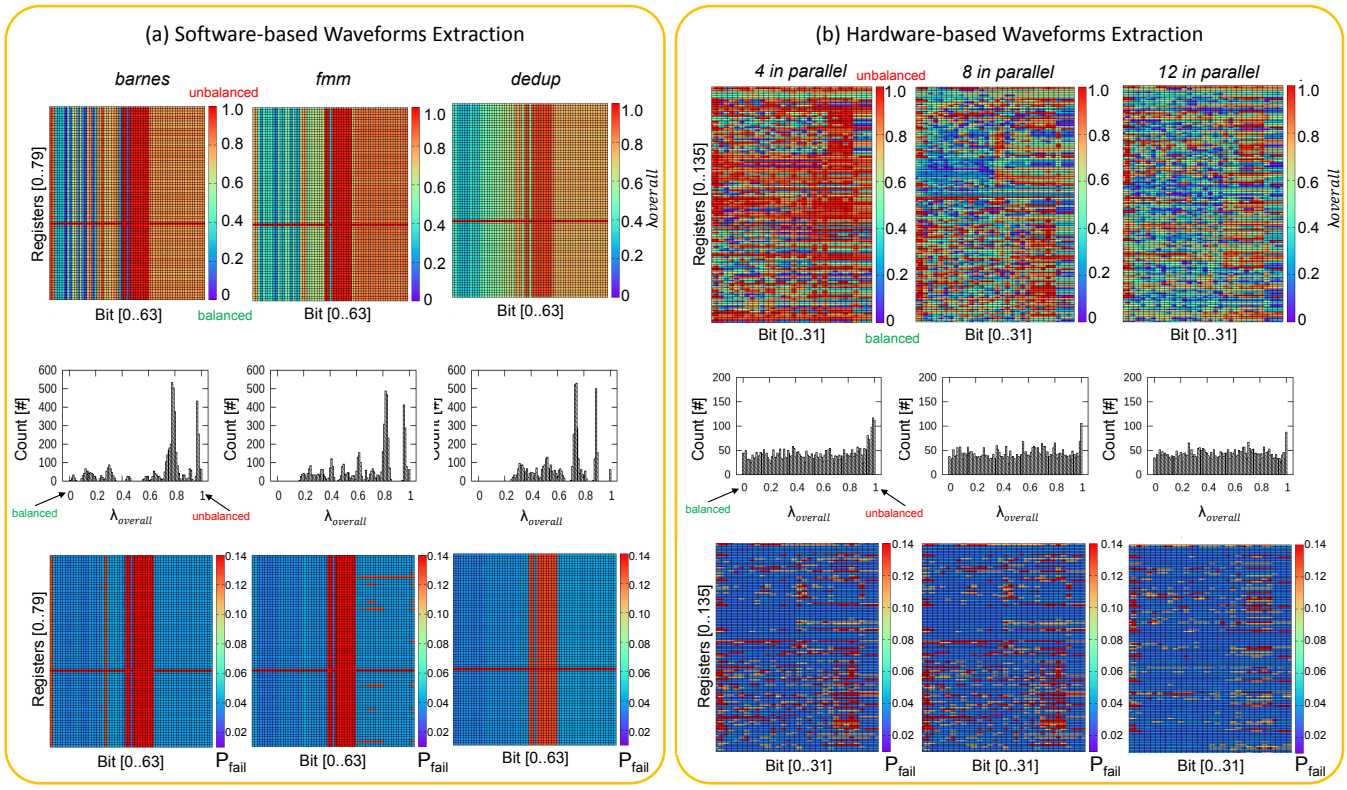

Fig. 9: Examples of the analysis for the cases of (a) An individual application (from the PARSEC suite) running on top of an operating system (i.e. *superficial* workload) using our software-based approach. (b) Multiple parallel applications (from the Mibench suite) running on top of the operating system (*typical* workload) using our hardware-based approach. The top row shows the $\lambda_{overall}$-maps to highlight the spatial distribution, while the middle row shows $\lambda_{overall}$-histograms to quantify the voltage-stress in the register file SRAMs. The bottom row shows the resulting P_{fail}-maps which interpret how aging effects ultimately induce failures

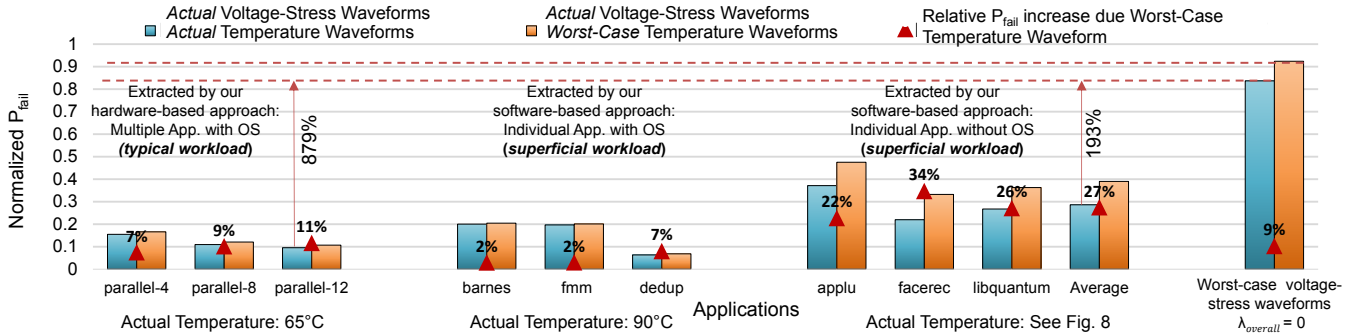

Fig. 10: Evaluation of the probability of failure for the employed applications grouped by their extraction approaches. The P_{fail} is normalized (i.e. expected number of broken SRAMs divided by the total number of register file SRAMs) for each architecture (Alpha, LEON3) for the sake of comparison. The operating system leads to smaller $\lambda_{overall}$ (i.e. more voltage-stress balancing in SRAMs) and hence lower P_{fail}. Running multiple applications in parallel noticeably makes register file SRAMs have more balanced voltage-stress, due to the interleaving between applications by the operating system scheduler. Considering the temperature/voltage-stress waveforms that the applications *actually* induce (as we propose) leads to accurately interpreting aging effects (i.e. without overestimation). As shown, considering the actual temperature waveform instead of the worst case (i.e. a constant temperature of $125^{\circ}C$) results in avoiding an overestimation of up to 33%. While considering worst-case voltage-stress waveforms (i.e. constant λ of 0) leads to a significant overestimation (up to 879%)

by the running applications. Then, our *compacting* process presented in Sections II-B compresses the waveforms to obtain $\lambda_{overall}$ for each single SRAM within the register file. Fig 8 highlights how different applications result in different maps of $\lambda_{overall}$. Furthermore, the spatial variation within the same map is considerable and thus an application can make SRAMs within the register file differently age. Similarly, we extract the impact of applications on the steady-state temperature of the register file. While Fig 8 presents the analysis of the case of running an individual application executed on bare metal through our software-based approach, Fig 9(a) demonstrates the analysis of the case of a individual application running on top of an operating system again based on our software-based approach. Whereas Fig 9(b),

the analysis of the case of multiple applications running in parallel on top of an operating system is based on our hardware approach. In addition to the $\lambda_{overall}$ maps, which help in capturing the SRAMs that are being under high stress, Fig 9 presents also the corresponding histograms to enable designers to quantitatively analyze the $\lambda_{overall}$ distribution.

Physical to application level: After extracting the temperature/voltage-stress waveforms, the physical aging models calculate the induced degradations and then our abstraction process discussed in Section II-E derives the corresponding probability of failure for each SRAM with the register file. We considered a typical safety margin of 10% (further details in Section II-E and [1]). The resulting P_{fail} maps for the register file SRAMs are presented in Figs (9, 8) and they show how aging effects actually manifest themselves at the system level. Finally, Fig 10 demonstrates the normalized probability of failure for each application along with a comparison with the worst-case scenario that comes from considering constant temperature/voltage-stress waveforms, as some of state-of-the-art do.

IV. Conclusion

We explored how the physical and application levels can be properly connected towards grasping when aging fails systems. We demonstrated how temperature and voltage-stress waveforms, originating from the application level, activate defects at the physical level which result in various aging effects through the abstraction levels up to the application level, where the ultimate impact of these effects take place. For that purpose, we presented software and hardware-based approaches to deal with varied kinds of scenarios. Our presented failure probability investigation enables the designer to grasp the aging effects as it interprets how transistors aging actually fail systems. Failure probability maps represent the spatial distribution of aging-induced failures within the processor component. This, in turn, provides designers with clear hints regarding the susceptible parts that need to be protected towards increasing the on-chip system reliability.

Acknowledgment

We would like to thank Christian List, Michael Skinder and Parthasarathy Rao for their valuable efforts in conducting experiments. The UAB group acknowledges the support of the Spanish MINECO and ERDF (TEC2013-45638-C3-1-R) and the Generalitat de Catalunya (2014SGR-384).

References

[1] H. Amrouch, V. M. van Santen, T. Ebi, V. Wenzel, and J. Henkel, "Towards interdependencies of aging mechanisms," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, ser. ICCAD, 2014, pp. 478–485.

[2] B. Kaczer, S. Mahato, V. de Almeida Camargo, M. Toledano-Luque, P. Roussel, T. Grasser, F. Catthoor, P. Dobrovolny, P. Zuber, G. Wirth, and G. Groeseneken, "Atomistic approach to variability of bias-temperature instability in circuit simulations," in *Reliability Physics Symposium (IRPS), IEEE International*, 2011, pp. XT.3.1–XT.3.5.

[3] W. Wang, V. Reddy, A. Krishnan, R. Vattikonda, S. Krishnan, and Y. Cao, "Compact modeling and simulation of circuit reliability for 65-nm CMOS technology," *Device and Materials Reliability, IEEE Transactions on*, vol. 7, no. 4, pp. 509–517, 2007.

[4] T. Grasser, P.-J. Wagner, H. Reisinger, T. Aichinger, G. Pobegen, M. Nelhiebel, and B. Kaczer, "Analytic modeling of the bias temperature instability using capture/emission time maps," in *Electron Devices Meeting (IEDM), IEEE International*, 2011, pp. 27.4.1–27.4.4.

[5] M. Toledano-Luque, B. Kaczer, J. Franco, P. Roussel, T. Grasser, and G. Groeseneken, "Defect-centric perspective of time-dependent BTI variability," *Microelectronics Reliability*, vol. 52, pp. 1883 – 1890, 2012.

[6] E. Donkoh, T. S. Ong, Y. N. Too, and P. Chiang, "Register file write data gating techniques and break-even analysis model," in *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design*, ser. ISLPED'12, 2012, pp. 149–154.

[7] H. Amrouch, T. Ebi, and J. Henkel, "Resi: Register-embedded self-immunity for reliability enhancement," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 33, no. 5, pp. 677–690, 2014.

[8] F. Mesa, M. Brown, J. Nayfach, and J. Renau, "Measuring power and temperature from real processors," *Parallel and Distributed Processing, IEEE International Symposium on*, pp. 1–5, 2008.

[9] H. Amrouch, T. Ebi, and J. Henkel, "Stress balancing to mitigate nbti effects in register files," in *Dependable Systems and Networks (DSN), 43rd Annual IEEE/IFIP International Conference on*, 2013, pp. 1–10.

[10] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood, "The Gem5 simulator," *SIGARCH Comput. Archit. News*, vol. 39, no. 2, pp. 1–7, 2011.

[11] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, "The mcpat framework for multicore and manycore architectures: Simultaneously modeling power, area, and timing," *ACM Trans. Archit. Code Optim.*, vol. 10, no. 1, pp. 5:1–5:29, 2013.

[12] M. R. Stan, K. Skadron, M. Barcella, W. Huang, K. Sankaranarayanan, and S. Velusamy, "Hotspot: a dynamic compact thermal model at the processorarchitecture level," *Microelectronics Journal*, vol. 34, pp. 1153–1165, 2003.

[13] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The PARSEC benchmark suite: Characterization and architectural implications," in *Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques*, 2008, pp. 72–81.

[14] J. L. Henning, "SPEC cpu2006 benchmark descriptions," *SIGARCH Comput. Archit. News*, vol. 34, no. 4, pp. 1–17, Sep. 2006.

[15] "LEON3," http://www.gaisler.com.

[16] H. Amrouch, T. Ebi, J. Schneider, S. Parameswaran, and J. Henkel, "Analyzing the thermal hotspots in FPGA-based embedded systems," in *Field Programmable Logic and Applications (FPL), 23rd International Conference on*, 2013, pp. 1–4.

[17] A. Bhattacharjee, G. Contreras, and M. Martonosi, "Full-system chip multiprocessor power evaluations using fpga-based emulation," in *Low Power Electronics and Design (ISLPED), ACM/IEEE International Symposium on*, 2008, pp. 335–340.

[18] M. Guthaus, J. Ringenberg, D. Ernst, T. Austin, T. Mudge, and R. Brown, "MiBench: A free, commercially representative embedded benchmark suite," in *Workload Characterization, 2001. WWC-4. 2001 IEEE International Workshop on*, 2001, pp. 3–14.

[19] M. Moras, J. Martin-Martinez, R. Rodriguez, M. Nafria, X. Aymerich, and E. Simoen, "Negative bias temperature instabilities induced in devices with millisecond anneal for ultra-shallow junctions," *Solid-State Electronics*, vol. 101, no. 0, pp. 131 – 136, 2014.

[20] J. Martin-Martinez, B. Kaczer, M. Toledano-Luque, R. Rodriguez, M. Nafria, X. Aymerich, and G. Groeseneken, "Probabilistic defect occupancy model for NBTI," in *Reliability Physics Symposium (IRPS), IEEE International*, 2011, pp. XT.4.1–XT.4.6.

[21] R. Fernandez, B. Kaczer, A. Nackaerts, S. Demuynck, R. Rodriguez, M. Nafria, and G. Groeseneken, "AC NBTI studied in the 1 hz – 2 ghz range on dedicated on-chip CMOS circuits," in *Electron Devices Meeting, 2006. IEDM '06. International*, 2006, pp. 1–4.

[22] J. Martin-Martinez, B. Kaczer, R. Degraeve, P. Roussel, R. Rodriguez, M. Nafria, X. Aymerich, B. Dierickx, and G. Groeseneken, "Circuit design-oriented stochastic piecewise modeling of the postbreakdown gate current in MOSFETs: Application to ring oscillators," *Device and Materials Reliability, IEEE Transactions on*, vol. 12, no. 1, pp. 78–85, 2012.