Leakage Models for High-Level Power Estimation

Domenik Helms, *Member, IEEE*, Reef Eilers, Malte Metzdorf, and Wolfgang Nebel, *Fellow, IEEE*

***Abstract*—Leakage currents are one major concern when designing recent CMOS devices, making design for leakage at all stages of the design process mandatory. Early leakage optimization requires early leakage prediction, and for electronic system level design, this means estimation capabilities at register transfer (RT) level or above. Existing models are very accurate, but slow [transistor level such as Berkeley Simulator (BSIM)], or the slightly faster gate level models (such as the Liberty library), disregard relevant parameters. We present RT level leakage macro models, which are faster than recent gate level models, while preserving the accuracy of the transistor level models to a great extent. An estimation framework is proposed, describing the subthreshold, gate, and junction leakage of recent technology devices. The models are characterized using BSIM compact models and a Monte Carlo process variation description. Each varying BSIM parameter can be described. As an example of use, channel length, oxide thickness, and channel doping are regarded together with the temperature, supply voltage and body voltage. The final macro model needs less than a hundred parameters to capture the leakage behavior of an entire RT component and is still analytically describing the dependence to the process parameters. Compared to SPICE + BSIM, a model prediction is computed up to a hundred times faster for large RT components, and is, depending on the analyzed technology, within 2.1% (for 16-nm LP)–6.8% (for 65-nm bulk) deviation over a wide range of operating conditions and process variation settings.**

***Index Terms*—Electronic design automation and methodology, leakage power, modeling, process variation, semiconductor device simulation, yield modeling.**

I. Introduction

BEFORE the 1990s, only performance and area used to be main concerns for electronic system design. The straightforward way of meeting these was a *simple scaling* as suggested by Moore's law.

Till then, dynamic power slowly started to become the third concern, leading to the first revolutions in system design, which was the introduction of CMOS and constant field scaling. Even though *simple scaling* of transistor dimensions was no longer sufficient for the dynamic power concern, it was still *happy scaling* under constant field constraints.

The new millennium brought static currents (further referred as leakage) as the fourth concern and with them the next revolution in system design resulting in the ending of the GHz race, and fundamentally new high-k and multigate devices. Suddenly, *happy scaling* became more of a problem and less of a solution. Thus additional adaptive design techniques such as power gating (PG), dynamic voltage and frequency scaling (DVFS), multiple power domains (MPDs) [1], and adaptive body biasing (ABB) are nowadays used to keep static power under control.

Those adaptive techniques are implemented at the device level, but all have to be controlled from a system level view. Designers of recent systems thus need to get leakage predictions for the decisions they have to meet (control of voltage, frequency and body potentials), already when they have to meet them, i.e., at the early system specification. This renders the need for a fast, yet accurate leakage prediction methodology such as the model presented here.

With the advent of leakage the process, voltage, and temperature variations (PVT variations) became an issue as leakage currents show an exponential dependency onto relevant, but hard to control physical and device parameters such as temperature and threshold voltage. These PVT variations turned out to be a completely new challenge in terms of variability aware design as well as for modeling. Over a decade after the first reports on leakage and variation within the very large-scale integration (VLSI) community, design for variation is usually still done by guard-banding due to the absence of appropriate variation aware design predictions at all levels of abstraction. Instead, recent modeling approaches are still based on Monte Carlo at device level and on corner cases [2] at the gate level.

A low power modeling standard focusing on abstraction, PVT independent modeling ensuring model continuity and interoperability is under preparation under the auspices of the IEEE P2416 [3]. This standard would focus on the usage of parameter dependent leakage models such as the one described in this paper. A leakage model, fitting the IEEE P2416 requirements should predict the leakage per component, while still parametrically regarding PVT variation. The highest modeling accuracy can only be achieved, if the spatial correlation of the variation is separated into a part which is constant per component (global variation and large-scale gradients) and a part which is assumed to be statistical per component (random variations and short-scale gradients).

PVT variations also have significant influence onto the system performance, and the circuit speed is correlated to the leakage via their parameter dependencies. Thus, a variation model has to be directly coupled to a variation dependent performance model like a statistical static timing analysis. As DVFS and ABB tradeoff performance for power, having a consistent timing prediction together with a leakage model

Manuscript received March 8, 2017; revised June 7, 2017; accepted September 20, 2017. Date of publication October 6, 2017; date of current version July 17, 2018. This paper was recommended by Associate Editor S. Hu. *(Corresponding author: Domenik Helms.)*

D. Helms, R. Eilers, and M. Metzdorf are with the Department of Transportation, OFFIS Institute, 26121 Oldenburg, Germany (e-mail: domenik.helms@offis.de; reef.eilers@offis.de; malte.metzdorf@offis.de).

W. Nebel is with the Department of Computer Science, Division of Embedded Hardware/Software Systems, University of Oldenburg, 26129 Oldenburg, Germany.

Digital Object Identifier 10.1109/TCAD.2017.2760519

This work is licensed under a Creative Commons Attribution 3.0 License. For more information, see http://creativecommons.org/licenses/by/3.0/

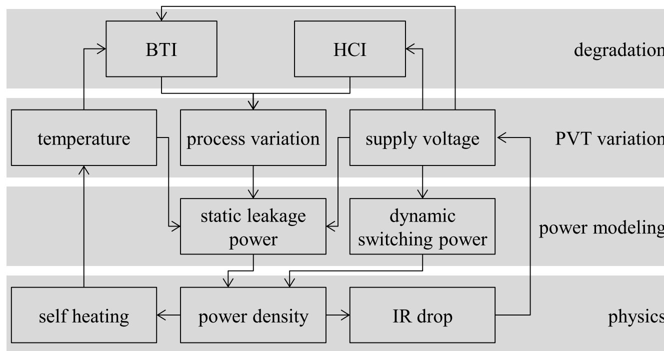

Fig. 1. Besides the well-known electro-thermal coupling and the supply voltage coupling via IR drops, degradation effects such as BTI and HCI introduce a cross coupling between static power and process variation.

is unavoidable for exploring the design space of variability aware design.

The tight positive correlation between (subthreshold) leakage current and temperature leads to the well-known effect of electro-thermal coupling. A higher temperature leads to increased leakage currents, which then dissipate into further thermal energy. Via IR drops, leakage and supply voltage are weakly correlated, too. This leads to a complex dependency between leakage, dynamic power, temperature, and supply voltage, which all have to be regarded accurately as reported in [4].

Recently, degradation (aging) effects gain importance, most prominently the bias temperature instability effect (BTI) and the hot carrier degradation (HCI). Depending on temperature and voltage level, the transistor's threshold voltage is increased by BTI over time and may also partially be reduced again if the operation conditions change [5]. HCI[1] results in a voltage dependent increase of the threshold voltage, mainly for nMOS devices [7]. As presented in Fig. 1, the BTI and HCI effect introduce a third leakage versus parameter feedback loop, introducing full leakage versus PVT coupling.

For a high-level leakage model, these couplings render the need for a tight integration into a multiphysics flow determining self-heating, IR drop, and degradation. Together with such a leakage model, suiting models for dynamic power and timing under PVT variation have to be developed, too.

Physics models, determining self-heating and IR drops from the local and spatial power density distribution have already been reported and will be detailed in the next section, where the state of the art is presented. There, we also refer to our earlier work on describing timing and dynamic power under PVT variation as well as on abstract BTI modeling.

In this paper, we focus on the leakage model, which we present in Section III as well as the integration into the overall flow, as detailed in Section IV. In Section V, we evaluate the accuracy of our model by comparing it to (Monte Carlo) SPICE and gate level simulations. Section VI, finally, concludes this paper.

[1]The physical mechanisms of HCI are still under scientific discussion, and it is not sure, that HCI is really caused by an injection [6]. Thus we carefully refer to is as hot carrier degradation, but use the common HCI acronym.

II. Related Work

One decade ago, leakage became known to the wide VLSI community. Early in 2003, Roy *et al.* [8] presented his famous overview over leakage physics and optimization. Later in 2003, Intel and IBM presented their leakage tutorial at the ICCAD conference, giving insight into device physics, process variation and optimization techniques [9]. Within the subsequent decade, several groups worked at modeling leakage, the influence of PVT variation and the usage of such models for design automation, supporting design for leakage and design for variability techniques.

A. Leakage Modeling

By introducing the Berkeley Simulator (BSIM) [10], [11], the predictive technology model (PTM) [12], and a set of reference MOSFET model cards ranging from 180 to 7 nm, and an automated model card generation methodology [13], Berkeley and Arizona State enabled evaluation of leakage estimation and optimization approaches without the need for expensive silicon production or intimate knowledge of confidential industrial technology data. Throughout this paper, the SPICE + BSIM simulation results are used as a baseline for all evaluation.

Already in 2000, Butts and Sohi [14] presented a full chip complexity-based leakage estimation methodology, separating leakage into transistor count n, supply V_{DD}, leakage per device I_{dev} and a design complexity metric k as

$$I_{\text{leak}} = k \cdot n \cdot V_{\text{DD}} \cdot I_{\text{dev}}. \tag{1}$$

This approach was then extended by Zhang *et al.* [15], resulting in the HotLeakage tool, by separating the transistor count, the complexity metric and the leakage per device, into an nMOS and a pMOS part (e.g., I_N and I_P), also adding an individual supply and temperature T dependence

$$I_{\text{leak}} = k_N \cdot n_N \cdot I_N(V_{\text{DD}}, T) + k_P \cdot n_P \cdot I_P(V_{\text{DD}}, T) \tag{2}$$

this paper basically presents an extension of this core idea.

Another extension of this fundamental work was presented by IBM in [16], introducing the concept of power contributors. It relies on the same principle of describing leakage as the multiple of elementary units, which can capture the leakage's dependence onto physical parameters. In comparison to the work presented here, Dhanwada *et al.* [16] offered a higher accuracy at the gate level for the cost of requiring full circuit gate level simulation as well as a few SPICE simulations for each model execution. Earlier IBM work [17] presented a sensitivity-based model which [18] and [19] extended, regarding intradie and interdie variation of the printed gate length. The characterization-based model can accurately predict the probability density function of the leakage distribution. Rao *et al.* [20] reported gate level state dependent leakage estimation by identifying all possible leakage states of a transistor. Instead, in [21], a static data-dependency analysis predicts the average (data independent) leakage with a general model using the leakage sensitivity to an arbitrary parameter.

In [22] and [23], thermally dependent leakage estimation was combined with a chip-wide temperature prediction, thus also regarding the electro-thermal back-coupling introduced by the subthreshold current's thermal dependence. Narendra *et al.* [24] analyzed the impact of threshold voltage variations in order to handle intradie process variations. A complete high-level leakage model was presented by Borkar *et al.* [25], regarding all PVT variations, thus all parameters except for the state.

Chen *et al.* [26] presented a subthreshold leakage estimation methodology for computing lower and upper bounds by regarding the stacking effect. Mukhopadhyay *et al.* [27] analyzed the leakage distribution for the one and two input gates reporting a substantial state dependency. There are many approaches at this level enabling accurate leakage current prediction if all relevant parameters are exactly known per device. But some parameters are not accurately predictable on the lower levels. The effect of parameter variations on leakage was analytically investigated in [28]. This approach analyzes the most important sources of leakage and predicts their distribution. In order to combine different parameters, an iterative approach is presented by Su *et al.* [4], accurately modeling dynamic power and leakage power by regarding the interaction between temperature, supply voltage, and power consumption. The authors introduce a thermal system model handling the electro-thermal coupling, as well as a supply grid model handling the electro-electro coupling introduced by IR drops.

1) IEEE Low Power Study Group: In mid-2014, two IEEE working groups have been formed. P2415 [29] is focusing on an energy oriented description of the software and hardware with a focus on power management techniques. It allows a description of the power relevant hardware blocks (memory, clock tree, IP components, etc.) and other impacts such as software activity or user inputs. P2416 [3] specifies a meta-model for IP-centric power estimation and optimization. It focusses on the description of the impact of PVT variation, dynamic power management techniques, or workload variation. As it is a meta-model it does not describe a modeling technique, but rather the properties and interfaces, a model should offer. We try to keep the model proposed here as close as possible to the idea of the yet to be defined meta-model.

Best to our knowledge, there is no other group, working at PVT and state dependent register transfer (RT) (or higher) level leakage. Except for the state dependency, all the bits and pieces of an all-in-one RT level leakage model exist, but have not been composed into a holistic model so far. In the evaluation section we will also discuss, that regarding state dependency—even though having only a limited impact on leakage—is mandatory to reduce the model errors to below 4% standard deviation. Thus, we are the first to combine all ideas into a very compact and mature model with estimation accuracy, comparable to a transistor level simulation.

B. Process, Temperature and Voltage Determination

In [17], subthreshold leakage variation under intradie variation is computed; regarding gate length, oxide thickness, and channel doping. A parameter sensitivity analysis enables estimation of the effect of a variation on the average leakage. This paper motivated the variation engine used here, focusing on interdie modeling, handling the effect of intradie variation as a technology property by an increased expectation value only.

Bhardwaj and Vrudhula [30] regarded the distribution of variations by presenting a statistical leakage and delay model on gate level, thus enabling optimization of the gate size to minimize leakage power. In [31], a methodology is presented, estimating the probability density function of the leakage current due to process parameter (ProPar) variation, enabling accurate yield prediction, by introduction of an intradie gradient model.

Modeling the temperature distribution $\Theta(\vec{r}, t)$ resulting from a given power consumption is in principle very easy if the special and temporal power distribution $P(\vec{r}, t)$, the system's heat capacitance per unit volume $C_V(\vec{r})$ and the system's thermal conductivity $\kappa(\vec{r})$ are known. First practical problems arise from the fact, that for the special power distribution, a floorplan has to be available and for the thermal system properties, the package geometry has to be known. Assuming, that a rough RT floorplan is available from tools like CompaSS [32] and that a package was roughly specified by the user by a flow like our [33], the temperature distribution results as

$$C_V\left(\vec{r}\right)\frac{\partial}{\partial t}\Theta\left(\vec{r}, t\right) = \vec{\nabla}\left(\kappa\left(\vec{r}\right)\vec{\nabla}\,\Theta\left(\vec{r}, t\right)\right) + P\left(\vec{r}, t\right). \quad (3)$$

This fundamental differential equation can be approximated using finite differences, which unfortunately turns out to be numerically far too complex for most practical problems. Thus, recent research focusses on determining fast approximate solutions to (3).

The most common approach is to separate the system into a low number of blocks, assuming perfect thermal conductivity inside each block. This results in a simple RC-network or even just a single low-pass translating power into temperature. Under certain symmetry assumptions,[2] $\Theta(\vec{r}, t)$ can also be computed by folding the power response function (Green's function) $\Theta_\delta(\vec{r}, t) = \Theta(P = P_0\delta(\vec{r}, t))/P_0$[3] with the power distribution

$$\Theta\left(\vec{r}, t\right) = \int d\varrho_0 \int d\varrho_1 \int d\tau P\left(\vec{\varrho}, \tau\right)\Theta_\delta\left(\vec{r} - \vec{\varrho}, t - \tau\right). \quad (4)$$

The Green's function-based thermal estimation (also referred to as power blurring or LUT-based thermal simulation) is subject to massive research for almost ten years. Recent research focusses on overcoming its limitations. In [34], the general concept for the static problem ($P(\vec{r}, t) = P(\vec{r})$) was initially presented, Hériz *et al.* [35] then introduced a description for the lateral chip boundaries. Park *et al.* [36] then added support for multilayer power input and multilayer thermal prediction. Ziabari *et al.* [37] developed a description for

[2] $C_V(\vec{r}) = C_V(r_2)$, $\kappa(\vec{r}) = \kappa(r_2)$, and $P(\vec{r}, t) = P(r_0, r_1, 0, t)$.

[3] Where $\delta(\vec{r}, t)$ is the 4-D Dirac function.

the full dynamic problem and Oh *et al.* [38] finally introduced approximation possibilities for inhomogeneities such as trough silicon vias.

As indicated in Fig. 1, the supply voltage has a strong impact onto dynamic power, static power, and degradation. All three show a strong, over-linear dependency toward the supply voltage level. For dynamic power, it is the well-known quadratic dependency of capacitance charging. For the static power, it is mainly caused by the drain induced barrier lowering effect, where a higher supply voltage linearly reduces the effective threshold voltage and thus exponentially increases subthreshold current [39]. For BTI, the capture times (the average times to activate a threshold voltage increasing trap under gate stress) almost show an exponential dependence onto the supply voltage [5]. Finally for HCI, the overall lifetime due to HCI exponentially depends on the per transistor drain-source current and thus onto the supply voltage [7]. Thus, a good understanding of the supply voltage distribution over the die is mandatory for power as well as aging prediction.

Among the effects, influencing the voltage level, reaching an individual area on the die, the IR drop has the strongest impact onto power and degradation. Assuming to have a static power dissipation, thus a static current demand due to static (leakage) power and a quasi-static current demand due to an average capacitance charging, the supply voltage drops over the supply grid's resistances and the central die area always sees a voltage reduction which is also typically rather stable from cycle to cycle. Any oscillations of the supply due to a rapid change in dynamic power can be neglected, as their over- and undershoot cancel out in first order in terms of back-influence onto the power consumption. In contrast, other effects lead to high frequency oscillations. Their influence onto power and degradation is of second order only, as over- and under-shooting of the voltage will partly cancel each other out. In [40], we propose a methodology, developing a voltage level and current density distribution from a given power distribution and supply grid topology.

C. Degradation Modeling

Both, BTI as well as HCI describe charges, trapped inside the gate oxide, causing a degradation of a device's threshold voltage. Recent models at (and below) the electrical level describe both effects by modeling the explicit [5] or statistical [41] occupation of these traps at time scales ranging from nanoseconds to years. Eilers *et al.* [42] accurately abstracted this trap occupation with only three degradation-parameters $\vec{d}$ per device. The long time degradation behavior under dynamical stress conditions such as varying temperatures or macroscopic active/idle periods can be described by a 6-D phase space, assigning new degradation-parameters $\vec{d}$ after a stress period T_{stress}, depending on the degradation parameters before this period as well as the average temperature θ_{avg}, voltage level V_{avg} and duty cycle x_{avg} within this stress period as described by

$$\vec{d}(t+\mathrm{T}_{\text{stress}}) = f\left(\vec{d}(t), \theta_{\text{avg}}, V_{\text{avg}}, x_{\text{avg}}\right). \quad (5)$$

As the stress period T_{stress} can be of order of seconds or even larger within reasonable error, a life-time evaluation of a device under degradation is enabled. In [43], we discuss the sources of inaccuracy from this abstraction and present a way to mitigate those.

If such a per device analysis is combined with a netlist-pruning, identifying potential critical paths as presented in [44], it enables describing the effect of degradation onto the all life timing behavior for entire RT components. In [45], we present the concept of the entire RT level, aging, and PVT induced timing degradation prediction flow.

III. Leakage Modeling

In this section, the requirements for an RT level leakage model, fitting into a multiphysics flow as described in Section II are defined and the model is described. From Fig. 1, it is obvious, that such a model should have temperature and supply voltage as model parameters and should also regard the effect of process variation.

Our model can handle each BSIM parameter as varying due to process variation with a user specified distribution, but is in practice limited to less than ten parameters for complexity reasons. After discussion with industrial applicants, we also reduced the description of local parameter variation[4] to a normal distribution around a mean value. The mean is determined by the global variation, and the standard deviation of local variation per parameter is a technology inherent property which is modeled only as an increased expectation value.

Guiding high-level design techniques is the ultimate purpose of our model, thus it has to regard all relevant high-level design for leakage techniques. Optimization techniques such as DVFS and MPD are supported by our model, simply as it has the supply voltage per RT component as an input parameter. In [46], we present models describing the transition cost as well as the remaining leakage when applying PG. Finally, ABB is recently regaining attention as it is well supported by recent FD-SOI technologies such as [47]. In order to support ABB, we thus add the body voltage as a further model parameter.

As an example of usage, in this paper we choose three ProPar: 1) channel length L_{ch}; 2) oxide thickness T_{ox}; and 3) channel doping N_{dep}. Thus we need to develop an easy to use leakage macro model, accurately describing the leakage of an entire RT component under variation of the ambient parameters (*AmbPar*) temperature, supply voltage, body voltage,[5] as well as the *ProPar* channel length, oxide thickness,[5] and channel doping.[5]

The model, fulfilling these harsh requirements is built in a bottom up way. A simple, yet accurate semianalytical model, described in Section III-B, captures the behavior of single transistors and small reference circuits toward all the model parameters. A gate level regression model, described in Section III-A, uses the transistor models to describe all gates from a technology library in all input states, still preserving the analytical parameter dependency from the transistor model

[4]Explicit local variation is fully supported in our initial work [48].

[5]Separately for nMOS and pMOS.

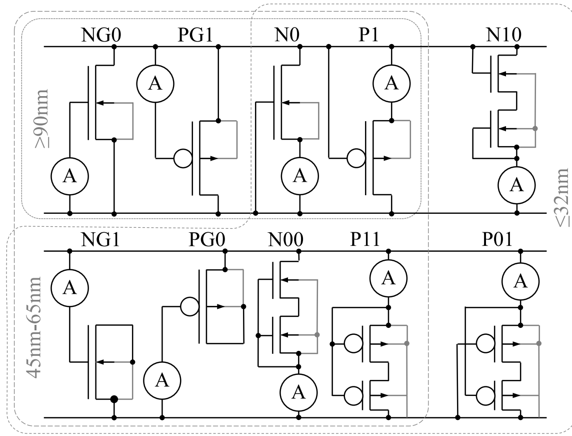

Fig. 2. Selection of reference circuits. In oldest technologies of 90 nm and above, only four references where needed, representing all relevant states sufficiently. For 45 and 65 nm, gate leakage only and stacking effect had to be supported, too. With high-*k* devices, the gate tunneling support could be removed, but the stacking effect had to be supported by further references.

but being further abstractable toward RT level. Finally, the RT model, presented in Section III-C, abstracts from the explicit data dependency at gate level resulting in real leakage macro models.

A. Library Characterization

The model was developed in several iterations [48]–[50], improving the accuracy and/or adapting the model to smaller node sizes. With the advent of high-*k* devices, further adaptations became necessary, as described below.

For each technology to be modeled, a small number of reference circuits (just consisting of one or two transistors, a current-meter, and specified voltage levels at all device terminals) has to be defined. This is the only model part that has to be adapted every two or three technology generations due to the rapid technological development.

For thick oxide SiNO bulk devices (90 nm and above), four reference circuits are needed. Each is referring to the current at the source terminal of an nMOS/pMOS transistor, when in a typical conducting/locking situation [see Fig. 2(top left)]. As these old devices mainly suffer from subthreshold leakage and a minor gate tunneling contribution, these four circuits represent all typical leakage conditions.

For thin oxide, high variability, SiNO bulk (45–65 nm) devices additional reference circuits have to be added in order to separate the description of gate and subthreshold leakage as well as to describe the stack effect [51]. In order to fully separate gate tunneling from subthreshold leakage, the current-meters for the gate tunneling are attached to the gate terminal. In order to also capture the impact of gate induced drain leakage (GIDL), the subthreshold references are moved from the source to the drain plus body terminal (see Fig. 2).

For high-*k* SOI (32 nm and below), GIDL as a body effect can be ignored. Gate tunneling only plays a minor role in most PVT corners, but not in all. For instance, the gate tunneling of a 16-nm PTM high-*k* metal gate device can be over twice as high as the subthreshold leakage (for high voltage, low temperature, and long channel) but is usually much lower (below 1% of the subthreshold voltage for low voltage, high temperature, and short channel). Instead, series of transistors show an interesting behavior atop of the traditional body effect, occurring in two locking transistors in sequence. For instance, the ratio between the subthreshold leakage of a N01 and a N10 stack (see Fig. 2) is over five times higher for high supply voltages, long transistor channels, and low temperatures, than in the opposite PVT corner. This renders the need for having again slightly updated reference circuits for these recent technologies. There are several options for a set of references, all performing almost equally in terms of average and maximum error. The most obvious one was replacing the NG0 and PG1 references in Fig. 2 by N10, and P10 (set up similarly to N00 and P11), thus replacing some gate tunneling description by better stack modeling. This set of references showed good average errors (0.19% std. dev.) and acceptable maximum errors (up to 4.34% for the AOI21 at 111 input). For higher modeling precision, two further references (N000 and P111) can be added, having some effect onto the model accuracy. This alternative version is also evaluated (see Tables III and IV).

The library characterization itself is then straight-forward. First, a list of N sets of representative PVT conditions $\underline{P} \in \mathbb{R}^{N\times P}$, $\vec{p}_p \in \mathbb{R}^{N\times 1}$ is defined, where $p < P$ denotes the pth variation parameter. This list does not necessarily have to be regular, or to cover all occurring conditions. Even if one or more of the PVT parameters never change in this list, the final model will be able to describe the leakage depending on these parameters. Parameters for nMOS and pMOS can be defined independently (e.g., for dopant concentration) or identically (e.g., for supply voltage). Table I details our choice of PVT variation ranges per technology.

Then, all gates $g \leq G$ in the library are simulated under all possible input patterns $s \leq S$ and for each of the PVT conditions. The total current flowing through this gate and into and out of the gate's inputs under static conditions is stored as the *gate leakage vector* $\underline{G}(\vec{p}) \in \mathbb{R}^{N\times G\cdot S}$, $\vec{g}_{g,s}(\vec{p}) \in \mathbb{R}^{N\times 1}$. The appropriate set of K reference circuits is implemented in SPICE and also simulated for the same PVT conditions $\underline{P}$, obtaining the *reference leakage vector* $\underline{R}(\underline{P}) \in \mathbb{R}^{N\times K}$, $\vec{r}_k(\underline{P}) \in \mathbb{R}^{N\times 1}$.

Finally, a linear parameter regression is used to represent each of the gate leakage vectors as a linear combination of the reference leakage vectors minimizing the quadratic error of the regression. The final library model $\vec{m}_{g,s} \in \mathbb{R}^{K\times 1}$ then simply results as

$$\vec{g}_{g,s} = \underline{R}\,\vec{m}_{g,s} \;\Rightarrow \vec{m}_{g,s} = \left(\underline{R}^T\underline{R}\right)^{-1}\underline{R}^T\ \vec{g}_{g,s} \tag{6}$$

$$\underline{R} = \begin{pmatrix} r_{1,1} & & \\ \vdots & \ldots & \vec{r}_K \\ r_{1,N} & & \end{pmatrix} \in \mathbb{R}^{N\times K}. \tag{7}$$

For a typical application scenario, the number of gates in a library is $G < 1000$ and the number of input pattern per gate is $S \leq 64$. $N = 1000$ PVT conditions was always enough for a meaningful regression and for all technologies analyzed

so far, $K \leq 8$ reference circuits were sufficient. Thus, an entire library characterization needs $N \cdot (G \cdot S + K) < 64$ Mio. static SPICE simulations which take some hours on a decent machine and produces exactly $G \cdot S \cdot K$ float values, which is just a few hundred kB of data.

B. Technology Abstraction

As presented in Section III-A, the leakage currents of all gates in a library under PVT variation can be accurately described by a linear combination of just eight reference circuits under the same PVT conditions. A simple sampling of the $\vec{r}_k(\vec{p})$ for a regular parameter space $\vec{p}$ followed by an interpolation is possible here, but impractical for two reasons.

On one hand, we intend to analytically include the impact of local process variation instead of having it as additional model parameters. Thus we need to have analytical expressions, at least for the *ProPar*, supporting numerical integration. On the other hand, the need for doing SPICE simulations for all gates and all references under all regular combinations of the *ProPar*, individually varying for pMOS and nMOS quickly leads to a state explosion limiting the number of *ProPar* to 2 or 3. Having an irregular $\vec{p}$ enables handling many *ProPar*, as not each combination of each of the parameters has to be built.

Thus, we try to find a simple, yet accurate analytical expression, describing the overall leakage current of the reference circuits, as presented in Fig. 2, under the assumption, that the *AmbPar* such as temperature and voltages are constant. Analysis of the according full equations from the BSIM manual [39] (which are certainly far too complex to be used themselves for our model) give good hints about the general structure, such expressions need to have to describe the leakage's analytical behavior with a minimal number of fitting parameters

$$\vec{r}_k\left(\vec{p}_{\text{pro}}, \vec{p}_{\text{amb}} = \text{const}\right) = \vec{r}_k\left(V_{\text{th}}, T_{\text{ox}}, N_{\text{dep}}\right) \approx \exp\left(\alpha_{0,k} + \sum_{i=1}^{3} \alpha_{i,k} {V_{\text{th}}}^{\beta_{i,1}/2} {T_{\text{ox}}}^{\beta_{i,2}/2} {N_{\text{dep}}}^{\beta_{i,3}/2}\right) \quad (8)$$

or more general with $\vec{p}_{k,j}$ being the jth PVT parameter of the kth PVT condition in vector $\vec{p}_{\text{pro}}$

$$\vec{r}_k\left(\vec{p}_{\text{pro}}\right) \approx \exp\left(\alpha_{0,k} + \sum_{i=1}^{3} \prod_{j=1}^{p} \alpha_{i,k} \vec{p}_{k,j}^{\;\beta_{i,j}/2}\right). \quad (9)$$

The final RT model relies on the fact, that the $\beta_{i,j}$ can be assumed to be constant for all temperatures and voltages and identical for all reference circuits, but vary only with the technology node. Table II analyzes the consequences of this assumption. The $\beta_{i,j}$ are restricted to halve integers for empirical reasons (see below). After defining the optimal set of $\beta_{i,j}$, as described below, the $\alpha_{i,k}$ are then determined by linear parameter regression for each reference k and for a regular grid of combinations of the *AmbPar* and finally stored in a table

$$\alpha_{i,k} = \alpha_{i,k}(\text{tech}, \text{ref}, \theta, V_{\text{DD}}, V_{\text{BB}}), \ \ \alpha_{i,k} \in \mathbb{R} \quad (10)$$

$$\beta_{i,j} = \beta_{i,j}(\text{tech}), \beta_{i,j} \in \mathbb{Z}. \quad (11)$$

For each given technology, the $\beta_{i,j}$ are obtained by a heuristic search (tabu search) through the 9-D parameter space $\vec{\beta} \in \mathbb{Z}^9$. The target function to be reduced is defined by optimizing $\alpha_{i,k}(\vec{\beta})$ and then taking the mean square error between the $\vec{r}_k$ from simulation and the model according to (8). Table II reports the $\beta_{i,j}$ values found, as well as the standard deviation, which was below 4% for all technologies analyzed over more than five orders of magnitude of leakage currents to be modeled.

The empirical reason for taking half-integer exponents in (8) is a compromise between the mathematical structure of the leakage currents (containing square roots and reciprocals in the exponent) and limitations of the search heuristic, needing a low-dimensional search space with defined discrete steps. We also tried replacing $\beta_{i,j}/2$ in (8) by $\beta_{i,j}/4$, thus going closer to a posynomial interpolation. This will in fact slightly reduce the error for the cost of a higher heuristic search complexity. Halve integers seemed to be a good compromise between accuracy and complexity.

The assumption of (10) directly implies that all nMOS and all pMOS transistors within an RT component have only one threshold voltage, each. When using different threshold voltages for optimization purposes (e.g., LVT on the critical path and HVT else), the gates of a component have to be split after logic simulation into groups with similar threshold voltage. Eventually, each group has to be modeled as an individual component leading to an individual set of $\beta_{i,j}$. Table II shows, that such a separation is necessary. In general, the $\beta_{i,j}$ for the HP (low threshold voltage) and LP (high threshold voltage) are completely independent (even though some tend to show some similarities).

C. Architecture Abstraction

For the final abstraction step, toward a real RT macro model we need to assume, that the structure (i.e., RT component) to be modeled is focused to a local die area. If the gates are close enough, it is valid to assume, that all gates see the same voltages and temperatures as well as the same global variation and intradie variation. In other words, we need to assume, that all gates have the same PVT state, except for a per gate normally distributed process variation of the *ProPar*, stemming from locally uncorrelated parameter variations.

Under this locality assumption, the $\alpha_{i,k}$ for each of the references k is identical for each gate of the RT component, resulting in

$$I_{\text{leak,RT}}\left(\text{tech}, \text{ref}, \theta, V_{\text{DD}}, V_{\text{BB}}, V_{\text{th}}, T_{\text{ox}}, N_{\text{dep}}\right) = \sum_{g=0}^{G} I_g\left(\text{tech}, \text{ref}, \theta, V_{\text{DD}}, V_{\text{BB}}, V_{\text{th}}, T_{\text{ox}}, N_{\text{dep}}\right) \quad (12)$$

$$= \sum_{g=0}^{G}\sum_{k=0}^{7} \vec{m}_{k,g}\, \vec{r}_k(\alpha_{i,k}, \beta_{i,j})$$
$$= \sum_{k=0}^{7} \vec{r}_k(\alpha_{i,k}, \beta_{i,j}) \sum_{g=0}^{G} \vec{m}_{k,g}$$
$$= \sum_{k=0}^{7} \vec{r}_k(\alpha_{i,k}, \beta_{i,j})\, \vec{M}_k(\alpha_{i,k}, \beta_{i,j})$$
$$= \vec{r} \cdot \vec{M} \tag{13}$$
$$\vec{M}_k := \sum_{g=0}^{G} \vec{m}_{k,g}. \tag{14}$$

Note, that without the locality assumption, the $\alpha_{i,k}$ would depend on the gate, thus would be $\alpha_{i,k,g}$, thus prohibiting to switch the sums and to lump all $\vec{m}_{k,g}$ into a single $\vec{M}_k$ per component. The $\vec{M}_k$ contains the sum of all per gate scaling parameters for the entire RT component.

The overall model now looks as follows: for a given input vector at the component's inputs, the state of all gates can be determined, thus the $\vec{m}_{k,g}$ are all known (g codes the gate and state of all the component's gates) and can be summed up to an $\vec{M} \in \mathbb{R}^8$. $\vec{M}_k$ is not depending on the PVT variation at all. Instead, the dependency toward the *AmbPar* is implicitly described by the $\alpha_{i,k}$ and the dependency toward the *ProPar* is explicitly described by the $\vec{r}_k$ according to (8). The RT model can be used as follows.

Step 1: Given the technology under analysis, the $\beta_{i,j}$ are read from a precharacterized table.

Step 2: Given the *AmbPar* as mean (or static) values for the entire RT component to be modeled, the $\alpha_{i,k}$ can be read from the table. If the *AmbPar* parameters do not exactly match the tabulated values, the table entries over and under the specific values are read for a later (multi)linear interpolation.

Step 3: Given the *ProPar* as a normal distribution with mean and variance, the $\vec{r}_k$ can be computed using the mean values in (8), then adding a numerical integration to represent the local variation. If an interpolation is required from step 2, step 3 is repeated for all AmbPar table values and finally, a multilinear interpolation is done to result in the final $\vec{r}_k$ values.

Step 4: The vector product of $\vec{M}_k$ and $\vec{r}_k$ gives the final leakage current.

1) Concerning Local Variations: As mentioned above, we regard local variations of the *ProPar* by a numerical integration. This part was shifted to the end of the model description to leave the modeling itself conceptually clear. It can be done as follows.

The total leakage of an RT component is obviously the sum of the leakage of all its gates. We assume, that each *ProPar* is normally distributed for all gates. As the leakage is usually not linearly depending on the *ProPar*, the sum of the leakages of n gates is not n times the leakage of one gate, but (on average) n times the expectation value of the per gate leakage. Thus, the switching of the sums in (12) is not perfectly correct, and should instead look as follows:

$$E\left(\vec{r}_k\right) = \lim_{G\to\infty} \frac{1}{G} \sum_{g=0}^{G} \vec{r}_k\left(V_{\text{th}}, T_{\text{ox}}, N_{\text{dep}}\right)$$
$$= \int_{-\infty}^{\infty} \mathrm{d}v\; p(v) \int_{-\infty}^{\infty} dt\; p(t) \int_{-\infty}^{\infty} dn\; p(n)\; \vec{r}_k(v,t,n) \tag{15}$$

where $p()$ is the probability density function of the normal distribution, which is defined by the mean *ProPar* value per component and its local variation. As we could not find a closed solution to the integral

$$\int_{-\infty}^{\infty} d\vec{p}\; \exp\left(-\sum_{j=1}^{p} \frac{\left(\vec{p}_j - \mu_{pj}\right)^2}{2\sigma_{pj}^2} + \alpha_{0,k} + \sum_{i=1}^{3}\prod_{j=1}^{p} \alpha_{i,k}\, \vec{p}_{k,j}^{\;\beta_{i,j}/2} \right) \tag{16}$$

we solve this numerically, once the μ_{pj} and σ_{pj} are known.

IV. Integration Into Flow

The methodology as described above is not restricted to a specific set of EDA tools. In fact, it can be smoothly integrated into different typical industrial design flows. As an example of use, we present the integration of the model into our flow, which eventually was used for the experimental assessment, as presented in Section V.

In order to set up RT level leakage models as described above, we assume having the following.

1) The BSIM model card of the technology under analysis.
2) Mean and variance of all ProPar to be regarded.
3) Liberty description of the library.
4) A SPICE netlist for each of the gates in our library.
5) A gate level Verilog description of all RT components to be modeled.

A. Model Characterization

We use a TCL script, interfacing with Synopsys HSPICE, measuring the leakage of small circuits under a large number of varying *ProPar*, within 3σ of the specified variation. For evaluation purposes, we chose 90 000 random and irregular parameter combinations, but for a typical characterization, less than 1000 sets should be enough.[6] Each gate of the library as well as all reference circuits are fed into this script. This step is done once for each combination of the *AmbPar*; for five temperatures, five supply voltages, and three body potentials,

[6] Exactly this step limits the number of individual process parameters that can be regarded. For a good modeling quality, each process parameter needs to be characterized with at least three different values—typically individually for pMOS and nMOS. Thus having defined three ProPar, for a good coverage, we already need $3^6 = 729$ sets. As mentioned above, there is no need for a complete (or even regular) coverage of all corner cases, still we need more and more sets with a growing number of ProPar.

thus 75 times in total. The gate simulations are stored as the $\vec{g}_{g,s}(\vec{p})$ and the references as the $\vec{r}_k(\vec{p})$.

Afterwards, the $\beta_{i,j}$ as defined in (8) are optimized by a tabu search, using the $\vec{r}_k(\vec{p})$ values only. We chose those $\beta_{i,j}$, minimizing the average standard deviation for all references. The nine integer $\beta_{i,j}$ are the first data, to be saved in the final model. The remainder (technology characterization raw data) is stored for further library and/or RT component abstraction, but is not needed for model application.

In order to reduce the final overall model error, we now replace the $\vec{r}_k(\vec{p})$ from simulation with computed values, using the $\beta_{i,j}$ and the *ProPar*, and then determine the $\vec{m}_{k,g}$ for each gate in each state and for each combination of the *AmbPar*. These 75 float values per gate per state (library characterization raw data) are also not needed in the final model, but are stored in case a new RT component has to be characterized.

Finally, we employ a logic simulation of the RT component to be modeled using Synopsys Design Compiler (DC) with 100 randomly chosen input pattern for the primary inputs of the RT component, all having the same signal probability. We repeat this step bitwise for each possible signal probability. For an RT component with B primary inputs, there are $(B+1)$ possible signal probabilities (from all zeros to all ones) and for each of the $100 \cdot (B+1)$ input vectors, we extract the state of each gate of the RT component, using a TCL script controlling the DC simulator.

Using the tabulated $\vec{g}_{g,s}$ values obtained from SPICE simulation, an almost exact prediction of the overall leakage per RT component can be obtained (referred to as gate level model in the assessment section). Instead, for the full data dependent model, we compute the $\vec{M}_k$ according to (13) for each of the $100 \cdot (B + 1)$ input vectors and average them to $(B+1)$ different $\vec{M}_{k,b}$, one vector per signal probability. These $8 \cdot (B + 1)$ float values are stored for the full (data dependent) model. If the separation of input probabilities is omitted, we obtain eight float values in total, resulting in the black box (data independent) model, as discussed in the next section.

B. Model Application

We developed the model for being used inside an all system multi physics assessment flow [52], implementing all the dependencies, indicated in Fig. 1. An entire digital embedded system with accurate timing data annotated is functionally simulated in SystemC, resulting in time-value pairs at all is subcomponent's inputs. Using our model, these values can be translated into a process variation dependent distribution of the per component leakage, initially assuming nominal temperature and voltages. Together with an input dependent switched capacitance-based dynamic power model, the overall power over time trace (as a process variation dependent distribution) can be obtained per component. A rough RT floor-planning translates this into a power density over time map (a distribution of maps due to process variation) and a rough package and power grid model [40] can be used to determine the process variation dependent distribution of temperature and voltage over time maps.

TABLE I
PARAMETER SELECTION

Technology	L_{ch} [nm]	T_{ox} [Å]	N_{dep} [10^{18}cm^{-3}]	V_{DD} [V]
16nm HP NFET	19.8-24.2	8.55-10.5	6.30-7.70	0.60-1.00
16nm HP PFET	19.8-24.2	9.00-11.0	4.95-6.05	0.60-1.00
16nm LP NFET	21.6-26.4	10.8-13.2	6.30-7.70	0.80-1.20
16nm LP PFET	21.6-26.4	11.0-13.4	3.96-4.84	0.80-1.20
22nm HP NFET	23.4-28.6	9.45-11.5	4.95-6.05	0.70-1.10
22nm HP PFET	23.4-28.6	9.90-12.1	3.96-4.84	0.70-1.10
22nm LP NFET	25.5-30.8	12.6-15.4	4.95-6.05	0.85-1.25
22nm LP PFET	25.5-30.8	12.6-15.4	3.96-4.84	0.85-1.25
32nm HP NFET	30.6-37.4	10.4-12.4	3.71-4.53	0.80-1.20
32nm HP PFET	30.6-37.4	10.8-13.2	2.76-3.38	0.80-1.20
32nm LP NFET	32.4-39.6	14.6-17.6	3.71-4.53	0.90-1.30
32nm LP PFET	32.4-39.6	14.6-17.8	2.76-3.38	0.90-1.30
45nm HP NFET	40.5-49.5	11.3-13.7	2.92-3.56	0.90-1.30
45nm HP PFET	40.5-49.5	11.7-14.3	2.20-2.68	0.90-1.30
45nm LP NFET	42.3-51.7	16.2-19.8	2.92-3.56	1.00-1.40
45nm LP PFET	42.3-51.7	16.4-20.0	2.20-2.68	1.00-1.40
65nm bulk NFET	58.5-71.5	17.0-20.8	2.29-2.79	1.00-1.40
65nm bulk PFET	58.5-71.5	17.6-21.4	1.68-2.06	1.00-1.40

Choice of PVT variation ranges in our simulations. The *ProPar* are reported above. From the *AmbPar*, only V_{DD} is adapted per technology, temperature always is $\Theta \in [300K, 400K]$ and the body voltage always is $V_{BB} \in [-0.2V, 0.2V]$.

Now, for each component, the average temperature and supply voltage over time is known and the body bias (if applicable) can be determined from the state of the power management. The leakage model is updating the initial power density prediction, which than will lead to new power density, temperature and voltage map for each variation instance.

V. EXPERIMENTAL ASSESSMENT

We evaluate our methodology, using nine predictive technologies, ranging from 16 to 65 nm. All experiments in this section are done for each of these technologies, assuming PVT variation ranges as presented in Table I. For each of these technologies, we define a set of 90 000 single PVT conditions, which is used for all our evaluations, presented below.

A. Accuracy of the Technology Abstraction

At first, the error, introduced by the analytic description of the technology abstraction is analyzed. Instead of setting up interpolation tables for the leakage currents of the reference circuits for all parameter combinations of the *AmbPar* and *ProPar*, (8) is characterized with just nine parameters. These $\beta_{i,j}$ only depend on the technology itself, and are even identical for all reference circuits. Afterwards, the $\alpha_{i,k}$ are characterized for each reference circuit k and various combinations of the *AmbPar*. The dependency toward the *ProPar* does no longer have to be stored in a table, but can be computed by (8).

The error introduced by this *technology abstraction* is presented in Table II. For all technologies, the same range of *ProPar* is chosen, as reported in Table I, resulting in over five orders of magnitude spread (ratio between highest and lowest leakage value in simulation) for the high performance (low V_{th}) technologies and almost three orders of magnitude for

TABLE II
TECHNOLOGY ABSTRACTION MODEL

Technology	$\beta_{1,j}$	$\beta_{2,j}$	$\beta_{3,j}$	Std	Spread
16nm HP	0/1/0	-1/3/1	-9/1/-1	1.96%	157k
16nm LP	0/1/0	-1/0/0	0/10/0	1.15%	450
22nm HP	0/1/0	0/3/1	-9/1/-1	2.84%	135k
22nm LP	0/-1/0	-1/8/0	-5/-2/1	1.15%	580
32nm HP	0/1/0	-4/0/-1	-9/1/-1	3.22%	241k
32nm LP	0/1/0	-2/10/0	-5/-1/1	1.94%	730
45nm HP	0/-1/0	-8/-1/-1	-10/0/-1	3.93%	152k
45nm LP	0/1/0	-4/-1/1	-3/10/1	2.12%	1150
65nm bulk	0/1/0	-4/0/-1	-10/0/-1	1.46%	380k

TABLE III
EVALUATION OF THE LIBRARY ABSTRACTION

Technology	Case	Gate	State	Avg%	Max%
16nm HP	worst	6T RAM	010	3.38	35.8
16nm HP	median	OAI22	1101	0.20	4.21
16nm HP	best	NOR3	100	0.07	4.94
16nm LP	worst	NOR3	001	0.16	2.83
16nm LP	median	OAI22	1110	0.03	0.50
16nm LP	best	NOR2	10	0.00	0.23
22nm HP	worst	NAN3	000	84.7	3746
22nm HP	median	MUX2	100	4.00	40.7
22nm HP	best	NOR2	01	0.06	3.13
22nm LP	worst	NOR3	011	0.64	13.1
22nm LP	median	OAI22	0010	0.05	2.84
22nm LP	best	NOR2	10	0.00	0.17
32nm HP	worst	NAN3	000	173	4755
32nm HP	median	NAN3	011	7.48	224
32nm HP	best	TMUX2	110	0.67	19.6
32nm LP	worst	NOR3	101	2.48	183
32nm LP	median	OAI21	010	0.19	12.6
32nm LP	best	OAI22	1001	0.05	11.2
45nm HP	worst	NOR3	111	52.7	1523
45nm HP	median	6T RAM	011	7.76	356
45nm HP	best	TMUX2	110	0.78	18.6
45nm LP	worst	TMUX2	000	7.19	122
45nm LP	median	OAI22	0010	0.40	29.0
45nm LP	best	NOR2	01	0.12	10.4
65nm bulk	worst	AOI22	1101	39.2	7596
65nm bulk	median	AOI22	1011	3.84	319
65nm bulk	best	NAN2	10	0.01	2.75
average				5.31	

Accuracy of the library abstraction: A linear combination of the references, according to Equation (6) is compared with SPICE simulations for a set of 143 cells. The table reports the average relative and maximum relative error for all parameter combinations, defined in Table I. For each technology, only the highest, median and lowest average error gate is reported.

the low power (high V_{th}) ones. The modeling error is thus also slightly higher (1.46%–3.93%) for low V_{th} and lower (1.15%–2.12%) for high V_{th} devices.

B. Accuracy of the Library Abstraction

To evaluate the accuracy of the library abstraction, a generic library, available in all analyzed technologies is used. This library is limited to a representative selection of cells, containing one inverter, NAN and nor gates with 2–4 inputs, an AOI21, OAI21, AOI22, and OAI22 cell as well as a two-input multiplexer once as strict CMOS and once using transmission gates. Finally, there are two sequential cells, a regular 6T SRAM cell and a D-latch.

Table III presents the standard deviation of the library abstraction, when describing the leakage of entire gates as a linear combination of the reference circuits (see Fig. 2). For the sake of comparability (and in contrast to the model description in Section III), the same set of eight references N0, P0, NG1, PG0, N00, P11, N10, and P01 is used for all technologies under analysis, which leads to slightly higher error values for the older (45 and 65 nm) technologies.

TABLE IV
LIBRARY ABSTRACTION USING ADDITIONAL REFERENCE CIRCUITS

Technology	Case	Gate	State	Avg%	Max%
16nm HP	worst	NAN4	0010	5.91	46.1
16nm HP	median	OAI22	0111	0.20	4.19
16nm HP	best	NAN4	0111	0.05	3.80
16nm LP	worst	NOR4	0001	0.25	5.42
16nm LP	median	OAI22	0101	0.03	0.71
16nm LP	best	NOR4	1110	0.00	0.07
22nm HP	worst	NAN4	0010	7.16	79.0
22nm HP	median	TMUX2	100	0.36	15.4
22nm HP	best	NOR2	01	0.06	3.13
22nm LP	worst	NOR4	0101	0.73	56.3
22nm LP	median	AOI22	1010	0.03	0.93
22nm LP	best	NAN4	0001	0.00	0.09
32nm HP	worst	NAN4	1111	12.6	463
32nm HP	median	OAI22	0110	1.23	40.5
32nm HP	best	NAN4	0111	0.13	2.73
32nm LP	worst	NOR4	0101	2.86	132
32nm LP	median	NAN3	111	0.10	18.6
32nm LP	best	NAN4	0001	0.02	4.91
45nm HP	worst	NAN4	1111	18.4	746
45nm HP	median	NAN3	001	1.36	52.3
45nm HP	best	NAN4	0000	0.10	4.14
45nm LP	worst	NOR4	0101	7.13	117
45nm LP	median	OAI22	1101	0.37	11.5
45nm LP	best	NOR4	1110	0.09	12.0
65nm bulk	worst	NOR3	100	39.6	3509
65nm bulk	median	OAI22	1111	4.89	93.8
65nm bulk	best	NAN2	10	0.01	2.74
average				1.78	

Repetition of the analysis from Table III, but with two additional reference circuits (3 locking N/P MOS in series).

For the low power technologies, showing only three orders of magnitude leakage variation with PVT variation, the average error over all gates in all states and in all PVT conditions is far below 1%. For the high performance technologies with their large spread in leakage, most gates still show average errors below 10% but with some exceptions such as the NAN3 gate in state 000 in 32-nm high performance, which shows an average error of 173% with a maximum error (worst deviation for a single PVT state) of 4755% (almost a factor 50 off for the worst PVT case).

Adding larger stacks of locking transistors to the set of references can drastically reduce the error here, but for the cost of a more complex model. As the *per-gate* model is just an intermediate step toward the model application at RTL; and as these large errors occur only rarely and are almost not affecting the RTL accuracy, we chose to accept these errors here. Nevertheless, Table IV summarizes the errors resulting from adding three sequential locking transistors to the reference circuits (N000 and P111 in the notation of Fig. 2).

Addition of two references to a regression should never make its error larger. Nevertheless, for some technologies, the average error rose. This artifact is not in contradiction to the theory of regression, instead, it illustrates a general

TABLE V
EVALUATION OF THE GATE LEVEL MODEL

Bench-mark	16nm HP	16nm LP	22nm HP	22nm LP	32nm HP	32nm LP	45nm HP	45nm LP	65nm bulk	ttl
c17	0.12/3.54	0.01/0.40	1.88/35.7	0.02/1.32	3.21/64.9	0.08/12.4	2.70/54.1	0.20/11.0	4.78/165	1.44/165
c432	0.08/2.80	0.02/0.41	1.41/20.4	0.03/1.40	2.50/50.9	0.11/11.5	2.27/24.0	0.27/10.3	3.27/60.7	1.11/60.7
c499	0.08/2.46	0.01/0.27	1.62/17.5	0.01/0.88	2.58/46.8	0.07/10.4	2.20/20.2	0.18/9.26	2.61/52.6	1.04/52.6
c880	0.09/2.55	0.01/0.32	1.58/19.3	0.02/1.14	2.62/46.9	0.08/10.7	2.30/20.5	0.21/9.59	3.88/74.3	1.20/74.3
c1355	0.07/2.32	0.01/0.27	1.25/16.2	0.01/0.97	2.09/44.3	0.07/10.8	1.84/19.1	0.20/9.85	2.52/43.7	0.90/44.3
c1908	0.08/2.30	0.01/0.28	1.41/19.1	0.01/1.12	2.29/50.5	0.07/10.6	1.98/21.9	0.19/9.59	2.52/53.4	0.95/53.4
c2670	0.08/2.55	0.01/0.33	1.71/19.1	0.02/1.20	2.78/46.2	0.09/10.4	2.28/20.1	0.21/9.34	3.10/58.5	1.14/58.5
c3540	0.09/2.43	0.01/0.32	1.69/18.9	0.02/1.20	2.73/46.1	0.09/10.7	2.37/20.2	0.23/9.59	3.51/68.8	1.19/68.8
c5315	0.09/2.45	0.01/0.31	1.80/20.8	0.02/1.29	2.83/48.0	0.08/10.7	2.25/21.0	0.22/9.58	3.20/70.2	1.17/70.2
c6288	0.09/2.46	0.01/0.34	1.20/17.0	0.01/0.91	2.24/36.7	0.07/10.8	2.25/20.3	0.20/9.93	6.09/121	1.35/121
ttl	0.09/3.54	0.01/0.41	1.56/35.7	0.02/1.40	2.59/64.9	0.08/12.4	2.24/54.1	0.21/11.0	3.55/165	1.15/165

Average relative error and maximum relative error (both in %) for the combinational ISCAS circuits considering 90,000 PVT variation states as presented in Table I. The gate level model has information about the input states of all gates. The average over all relative errors and the maximum worst case error per technology as well as per benchmark are reported under ttl.

TABLE VI
EVALUATION OF THE RT LEVEL BLACK BOX MODEL WITHOUT DATA ABSTRACTION

Bench-mark	16nm HP	16nm LP	22nm HP	22nm LP	32nm HP	32nm LP	45nm HP	45nm LP	65nm bulk	ttl
c17	14.0/83.1	9.66/69.2	16.5/106	10.0/88.4	18.3/137	10.3/82.2	17.2/143	11.4/76.7	21.7/141	14.3/143
c432	3.37/26.1	2.88/18.1	6.38/39.0	2.94/18.0	7.09/43.2	3.04/25.4	6.42/38.2	3.42/27.7	9.09/51.6	4.96/51.6
c499	1.71/14.5	1.75/12.2	3.46/27.8	1.80/12.9	4.69/45.5	1.69/12.8	4.24/32.7	1.63/15.3	5.63/45.4	2.96/45.5
c880	2.11/16.6	2.06/13.9	4.08/25.8	2.18/15.6	5.08/41.8	2.07/15.3	4.54/26.5	2.20/16.6	6.83/67.1	3.46/67.1
c1355	1.99/20.4	1.75/8.39	4.64/33.4	1.74/10.1	5.34/36.3	2.01/16.4	4.82/32.5	2.35/17.0	6.84/41.4	3.50/41.4
c1908	2.14/26.1	2.20/10.7	4.39/37.6	2.14/14.0	5.06/40.1	2.37/23.7	4.49/35.4	2.65/25.0	6.16/40.8	3.51/40.8
c2670	1.62/9.86	1.56/7.27	3.37/20.7	1.54/7.46	4.52/42.8	1.53/12.5	3.95/24.4	1.62/12.0	5.40/46.0	2.79/46.0
c3540	1.68/11.2	1.70/8.28	3.53/21.0	1.75/8.98	4.54/43.9	1.79/12.5	4.07/24.0	1.94/12.5	5.78/60.8	2.98/60.8
c5315	2.11/19.6	2.00/14.9	3.68/25.5	2.12/16.8	5.02/47.9	2.04/16.2	4.54/38.7	2.08/15.8	6.06/55.1	3.29/55.1
c6288	1.38/9.12	1.42/10.0	2.48/17.1	1.47/9.96	3.40/31.9	1.46/13.6	3.16/21.2	1.53/15.3	7.05/114	2.59/114
ttl	3.21/83.1	2.70/69.2	5.25/106	2.77/88.4	6.30/137	2.83/82.2	5.74/143	3.09/76.7	8.06/141	4.44/143

Average relative error and maximum relative error as in Table V, but for a pure black box model without any data abstraction. This model describes leakage by a single PVT dependent, but input signal independent average value.

problem with error measures for leakage models and justifies our initial selection of eight references. Linear regression optimizes for minimal quadratic error, but a minimal quadratic error is not a perfectly suited error measure for a leakage model, having to predict currents over several orders of magnitude. Instead a good relative accuracy is needed. This is why we report the average relative error instead of the standard deviation

$$\text{Avg\%} := \frac{1}{N}\sum_{i=1}^{N} \text{model}_i/\text{simulation}_i \tag{17}$$

and this is also why the addition of the N000 and P111 references are cosmetic (to convince the reader, that the method is accurate enough), rather than required by the final model.

C. Accuracy of the Architecture Abstraction

The model (with eight reference circuits) is applied to the set of ISCAS85 benchmarks [53], to assess the data abstraction, as well as the overall model accuracy to be expected at the RT level. We intended to use SPICE as a baseline for all levels of abstraction. Unfortunately, for the largest benchmark, the HSPICE simulation did not converge at all. So instead, as a baseline, the component is simulated at gate level first, determining the logic level of each gate. Eventually, a separate SPICE simulation for each gate is done and the currents are accumulated. When accounting for gate tunneling correctly (as indicated in Fig. 2), this leads to almost no deviation.

Table V presents the average relative error as well as the maximum relative error, when computing the overall leakage using the data dependent RT level model in contrast to the gate level model. For the high performance technologies (top half of Table V) with their five orders of magnitude PVT variation, the model has an average relative error of 2.09% and a worst case error of 116%. That means that for the worst of the 90 000 PVT conditions analyzed, the 65-nm bulk technology shows a 116% deviation (a factor 2.16) between the *as good as SPICE* baseline and our model. The low power technologies (bottom half of Table V) show only three orders of magnitude leakage variation with PVT. The average relative error there is 0.08% and no single prediction is more than 11.1% off.

The same RT level evaluation was also done, using the ten references (the regular eight plus the N000 and P111), analyzed in Table IV. While the two additional references have a large impact onto the gate level results, the improvement at the RT level is far lower. Using ten references, the average error drops from 2.09% to 1.20% for the high performance and from 0.08% to 0.05% for the low power technologies.

TABLE VII
EVALUATION OF THE RT LEVEL BLACK BOX MODEL WITH DATA ABSTRACTION

Bench-mark	16nm HP	16nm LP	22nm HP	22nm LP	32nm HP	32nm LP	45nm HP	45nm LP	65nm bulk	ttl
c17	9.60/41.0	6.78/32.3	12.6/45.6	6.46/37.8	14.3/90.6	7.01/43.0	13.6/104	7.96/46.6	17.9/162	10.7/162
c432	2.93/24.2	2.54/15.9	5.67/35.3	2.55/16.0	6.33/45.8	2.67/22.7	5.65/34.0	2.99/24.8	8.14/53.3	4.39/53.3
c499	1.41/15.4	1.38/8.27	2.65/20.6	1.34/8.50	3.84/47.0	1.30/12.2	3.41/34.0	1.28/11.7	4.33/46.6	2.33/47.0
c880	1.78/16.3	1.62/7.95	3.44/24.3	1.67/8.48	4.51/46.9	1.62/14.3	4.06/26.0	1.74/15.1	6.13/69.0	2.95/69.0
c1355	1.45/12.8	1.39/7.45	2.67/21.3	1.38/7.73	3.59/43.8	1.38/12.7	3.24/21.2	1.42/12.0	4.40/40.0	2.32/43.8
c1908	1.95/20.3	1.91/10.3	3.91/30.2	1.88/11.1	4.59/43.3	2.02/17.9	4.07/32.1	2.26/18.8	5.51/46.1	3.12/46.1
c2670	1.27/8.83	1.29/8.06	2.89/19.9	1.26/8.00	3.98/44.9	1.27/11.8	3.40/23.4	1.34/11.4	4.68/56.4	2.38/56.4
c3540	1.47/9.07	1.50/8.07	3.10/19.9	1.52/8.69	4.14/44.9	1.55/12.5	3.67/22.3	1.64/12.6	5.21/64.5	2.64/64.5
c5315	1.77/16.1	1.63/10.3	3.23/22.5	1.71/11.6	4.53/47.8	1.66/12.9	4.02/33.9	1.71/14.0	5.40/60.4	2.85/60.4
c6288	0.79/8.08	0.87/4.40	1.92/17.0	0.84/4.86	2.97/36.7	0.87/11.8	2.86/20.9	0.95/12.0	6.71/121	2.09/121
ttl	2.44/41.0	2.09/32.3	4.20/45.6	2.06/37.8	5.28/90.6	2.14/43.0	4.80/104	2.33/46.6	6.84/162	3.58/162

Average relative error and maximum relative error as in Table V for the final proposed model. Regarding the signal probability at the primary component inputs significantly reduces the error. Average error values lie between the unrealistic white box model from table V and the straightforward model from Table VI.

TABLE VIII
SPEEDUP OF SIMULATION TIME

Benchmark	SPICE	gate vs. SPICE	RTL (input dep.) vs. gate	Input dep. vs. indep.
c17	350s	x2.18	x0.19	x1000 (# cycles)
c432	1294s	x3.53	x0.44	
c499	18534s	x23.48	x0.78	
c880	13033s	x16.57	x0.69	
c1355	16249s	x21.29	x0.65	
c1908	9626s	x15.11	x0.76	
c2670	19360s	x22.8	x0.84	
c3540	69572s	x85.53	x1.18	
c5315	72119s	x33.65	x2.13	
c6288	d.n.f.	3961.7s	x3.45	

Simulation times for all models discussed in this work. SPICE reports the time needed to do DC computations for 1000 consecutive input vectors using Synopsys HSPICE. C6288 did not converge at all. Gate vs. SPICE reports the relative speedup of the full per gate white-box model. RTL vs. gate presents the relative speedup versus the gate model. Values over 1 mean, that the RTL model is actually faster. Finally, as the input independent model only has to do a single model execution, it is almost exactly x1000 times faster than the input dependent version.

The worst case error is even significantly higher there, with 220% for high performance and 9.83% for low power. Adding data dependency to the model finally yields in an average error of below 4%, as presented in Table VII.

D. Model Execution Performance

For each of the benchmarks, and for each model discussed, a test-bench with 1000 consecutive input stimuli was computed. We chose such a large number in order to avoid to measure the setup times of the routines (such as DC or HSPICE boot time) only. Table VIII summarizes the simulation times for SPICE, followed by a relative speedup factor, when employing our gate model (as evaluated in Table V). While for small circuits, there is only a marginal speedup of factor 2, the gate level technique is scaling much better with the number of gates, reaching almost a factor of hundred for the largest components. The input dependent RT level model's execution time is not depending on the number of gates, thus resulting in a further speedup of factors 2 and 3 for large components. Finally, the input independent model's execution does neither depend on the number of gates, nor on the number of cycles. Instead, it only has to be reevaluated, once the *AmbPar* change.

VI. CONCLUSION

We presented an efficient and accurate per gate leakage model, which is describing the dependency toward process variations analytically and toward temperature, supply, and body voltage by interpolation. For the technologies analyzed, we used channel length, oxide thickness, and channel doping concentration as varying ProPar. For future technologies, introducing undoped channels and advanced geometries, further ProPar can be exchanged or added. The model is already useful at the gate level, enabling a fast PVT aware leakage prediction. The standard deviation versus SPICE was below 4% over more than five orders of magnitude and below 2.2%, if the leakage did vary only over three orders of magnitude. Best to our knowledge, there is no other model, taking all relevant leakage effects into account and resulting in a purely analytical model, which can predict leakage currents without needing any SPICE simulations for model application.

Using this base model, we could set up macro models, describing the leakage of an entire RT component without the need for gate level simulation details. For large RT components, the input independent black box RT model has just 1.2% higher error than the most accurate gate level model, while speeding up the model execution by more than a factor of 10 000. Adding input dependency to the RT model reduces this error to 0.7%, but requires to reevaluate the model for each clock cycle, reducing the speedup versus SPICE to less than a factor of hundred. Short of earlier versions of this paper [48]–[50], there is no other black box RT level leakage model.

REFERENCES

[1] A. Jarrar, *I Blew My Power Budget: Whom Should I Throw Under the Bus*? DAC Panel, Austin, TX, USA, 2013.

[2] N. Dhanwada *et al.*, “Efficient PVT independent abstraction of large IP blocks for hierarchical power analysis,” in *Proc. ICCAD*, San Jose, CA, USA, 2013, pp. 458–465.

[3] *Standard Project for Power Modeling to Enable System Level Analysis*, IEEE Standard P2416, Sep. 2014.

[4] H. Su, F. Liu, A. Devgan, E. Acar, and S. Nassif, "Full chip leakage-estimation considering power supply and temperature variations," presented at ISLPED, Seoul, South Korea, 2003, pp. 78–83.

[5] T. Grasser *et al.*, "The paradigm shift in understanding the bias temperature instability: From reaction–diffusion to switching oxide traps," *IEEE Trans. Electron Devices*, vol. ED-58, no. 11, pp. 3652–3666, Nov. 2011.

[6] S. Tyaginov *et al.*, "Impact of the carrier distribution function on hot-carrier degradation modeling," in *Proc. ESSDERC*, Helsinki, Finland, Sep. 2011, pp. 151–154.

[7] A. Bravaix *et al.*, "Hot-carrier acceleration factors for low power management in DC–AC stressed 40nm NMOS node at high temperature," in *Proc. IEEE IRPS*, Montreal, QC, Canada, Apr. 2009, pp. 531–548.

[8] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimad, "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits," *Proc. IEEE*, vol. 91, no. 2, pp. 305–327, Feb. 2003.

[9] D. Blaauw, A. Devgan, and F. Najm, "Leakage power: Trends, analysis and avoidance," in *Proc. Asia South Pac. Design Autom. Conf. (ASP DAC)*, vol. 1. Shanghai, China, 2005, p. 2. [Online]. Available: http://ieeexplore.ieee.org/document/1466116/

[10] Z.-H. Liu *et al.*, "Threshold voltage model for deep-submicrometer MOSFETs," *IEEE Trans. Electron Devices*, vol. 40, no. 1, pp. 86–95, Jan. 1993.

[11] C. Hu, "BSIM model for circuit design using advanced technologies," in *Proc. VLSI Circuits*, Kyoto, Japan, 2001, pp. 5–10.

[12] Y. Cheng *et al.*, "An investigation on the robustness, accuracy and simulation performance of a physics-based deep-submicronmeter BSIM model for analog/digital circuit simulation," in *Proc. CICC*, San Diego, CA, USA, 1996, pp. 321–324.

[13] Y. Cao, M. Orshansky, T. Sato, D. Sylvester, and C. Hu, "Spice up your MOSFET modelling," *IEEE Circuits Devices Mag.*, vol. 19, no. 4, pp. 17–23, Jul. 2003.

[14] J. A. Butts and G. S. Sohi, "A static power model for architects," in *Proc. MICRO*, Monterey, CA, USA, 2000, pp. 191–201.

[15] Y. Zhang, D. Parikh, K. Sankaranarayanan, K. Skadron, and M. Stan, "HotLeakage: A temperature-aware model of subthreshold and gate leakage for architects," Dept. Elect. Comput. Eng., Univ. Virginia, Charlottesville, VA, USA, Tech. Rep. CS-2003-05, 2003.

[16] N. Dhanwada, D. Hathaway, J. Frenkil, W. R. Davis, and H. Demircioglu, "Leakage power contributor modeling," *IEEE Des. Test Comput.*, vol. 29, no. 2, pp. 71–78, Apr. 2012.

[17] A. Srivastava, R. Bai, D. Blaauw, and D. Sylvester, "Modeling and analysis of leakage power considering within-die process variations," in *Proc. ISLPED*, Monterey, CA, USA, 2002, pp. 64–67.

[18] R. Rao, A. Srivastava, D. Blaauw, and D. Sylvester, "Statistical estimation of leakage current considering inter- and intra-die process variation," in *Proc. ISLPED*, Seoul, South Korea, Aug. 2003, pp. 84–89.

[19] R. R. Rao, D. Blaauw, D. Sylvester, and A. Devgan, "Modeling and analysis of parametric yield under power and performance constraints," *IEEE Des. Test Comput.*, vol. 22, no. 4, pp. 376–385, Jul./Aug. 2005.

[20] R. M. Rao, J. L. Burns, A. Devgan, and R. B. Brown, "Efficient techniques for gate leakage estimation," in *Proc. ISLPED*, Seoul, South Korea, Aug. 2003, pp. 100–103.

[21] E. Acar *et al.*, "Leakage and leakage sensitivity computation for combinational circuits," in *Proc. ISLPED*, Seoul, South Korea, 2003, pp. 96–99.

[22] W. Liao, F. Li, and L. He, "Microarchitecture level power and thermal simulation considering temperature dependent leakage model," in *Proc. ISLPED*, Seoul, South Korea, 2003, pp. 211–216.

[23] K. Banerjee, S.-C. Lin, A. Keshavarzi, S. Narendra, and V. De, "A self-consistent junction temperature estimation methodology for nanometer scale ICs with implications for performance and thermal management," in *Proc. IEEE Int. Electron Devices Meeting*, Washington, DC, USA, 2003, pp. 36.7.1–36.7.4.

[24] S. Narendra, V. De, S. Borkar, D. Antoniadis, and A. Chandrakasan, "Full-chip sub-threshold leakage power prediction model for sub-0.18μm CMOS," in *Proc. ISLPED*, Monterey, CA, USA, 2002, pp. 19–23.

[25] S. Borkar *et al.*, "Parameter variations and impact on circuits and microarchitecture," in *Proc. DAC*, Anaheim, CA, USA, 2003, pp. 338–342.

[26] Z. Chen, M. Johnson, L. Wei, and K. Roy, "Estimation of standby leakage power in CMOS circuit considering accurate modeling of transistor stacks," in *Proc. ISLPED*, Monterey, CA, USA, 1998, pp. 239–244.

[27] S. Mukhopadhyay, A. Raychowdhury, and K. Roy, "Accurate estimation of total leakage current in scaled CMOS logic circuits based on compact current modeling," in *Proc. DAC*, Anaheim, CA, USA, 2003, pp. 169–174.

[28] S. Mukhopadhyay and K. Roy, "Modeling and estimation of total leakage current in nano-scaled-CMOS devices considering the effect of parameter variation," in *Proc. ISLPED*, Seoul, South Korea, 2003, pp. 172–175.

[29] *Standard Project for Unified Hardware Abstraction and Layer for Energy Proportional Electronic Systems*, IEEE Standard P2415, Sep. 2014.

[30] S. Bhardwaj and S. B. K. Vrudhula, "Leakage minimization of nano-scale circuits in the presence of systematic and random variations," in *Proc. DAC*, Anaheim, CA, USA, 2005, pp. 541–546.

[31] H. Chang and S. S. Sapatnekar, "Full-chip analysis of leakage power under process variations, including spatial correlations," in *Proc. DAC*, Anaheim, CA, USA, 2005, pp. 523–528.

[32] H. H. Chan and I. L. Markov, "Practical slicing and non-slicing block-packing without simulated annealing," in *Proc. GLSVLSI*, Boston, MA, USA, 2004, pp. 282–287.

[33] S. Rosinger, M. Metzdorf, D. Helms, and W. Nebel, "Behavioral-level thermal- and aging-estimation flow," in *Proc. LATW*, 2011, pp. 1–6.

[34] T. Kemper, Y. Zhang, Z. Bian, and A. Shakouri, "Ultrafast temperature profile calculation in IC chips," in *Proc. THERMINIC*, Nice, France, Sep. 2006, pp. 133–137.

[35] V. M. Hériz, J.-H. Park, T. Kemper, S.-M. Kang, and A. Shakouri, "Method of images for the fast calculation of temperature distributions in packaged VLSI chips," in *Proc. THERMINIC*, Budapest, Hungary, Sep. 2007, pp. 18–25.

[36] J.-H. Park, A. Shakouri, and S.-M. Kang, "Fast thermal analysis of vertically integrated circuits (3-D ICs) using power blurring method," in *Proc. InterPACK*, San Francisco, CA, USA, Jul. 2009, pp. 701–707.

[37] A. Ziabari, E. K. Ardestani, J. Renau, and A. Shakouri, "Fast thermal simulators for architecture level integrated circuit design," in *Proc. SEMI THERM*, San Jose, CA, USA, Mar. 2011, pp. 70–75.

[38] D. Oh, C. C. P. Chen, and Y. H. Hu, "Efficient thermal simulation for 3-D IC with thermal through-silicon vias," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 31, no. 11, pp. 1767–1771, Nov. 2012.

[39] M. V. Dunga *et al.* (2006). *BSIM4.6.0 MOSFET Model—User's Manual*. [Online]. Available: http://www-device.eecs.berkeley.edu/bsim/Files/BSIM4/BSIM460/doc/BSIM460_Manual.pdf

[40] D. Helms, K. Hylla, and W. Nebel, "Hybrid logical-statistical simulation with thermal and IR-drop mapping for degradation and variation prediction," in *Proc. ISLPED*, San Francisco, CA, USA, 2009, pp. 33–38.

[41] H. Reisinger, T. Grasser, W. Gustin, and C. Schlünder, "The statistical analysis of individual defects constituting NBTI and its implications for modeling DC- and AC-stress," in *Proc. IRPS*, Anaheim, CA, USA, May 2010, pp. 7–15.

[42] R. Eilers, M. Metzdorf, D. Helms, and W. Nebel, "Efficient NBTI modeling technique considering recovery effects," in *Proc. ISLPED*, La Jolla, CA, USA, Aug. 2014, pp. 177–182.

[43] A. Unutulmaz *et al.*, "Analysis of NBTI effects on high frequency digital circuits," in *Proc. DATE*, Dresden, Germany, 2016, pp. 223–228.

[44] D. Lorenz, M. Barke, and U. Schlichtmann, "Monitoring of aging in integrated circuits by identifying possible critical paths," *Microelectron. Rel.*, vol. 54, nos. 6–7, pp. 1075–1082, Jun./Jul. 2014.

[45] N. Koppaetzky, M. Metzdorf, R. Eilers, D. Helms, and W. Nebel, "RT level timing modeling for aging prediction," in *Proc. DATE*, Dresden, Germany, 2016, pp. 297–300.

[46] S. Rosinger, D. Helms, and W. Nebel, "RTL power modeling and estimation of sleep transistor based power gating," in *Proc. PATMOS*, Gothenburg, Sweden, Sep. 2007, pp. 278–287.

[47] P. Magarshack, P. Flatresse, and G. Cesana, "UTBB FD-SOI: A process/design symbiosis for breakthrough energy-efficiency," in *Proc. DATE*, Grenoble, France, 2013, pp. 952–957.

[48] D. Helms, M. Hoyer, and W. Nebel, "Accurate PTV, state, and ABB aware RTL blackbox modeling of subthreshold, gate, and PN-junction leakage," in *Proc. PATMOS*, Montpellier, France, Sep. 2006, pp. 56–65.

[49] D. Helms, M. Hoyer, S. Rosinger, and W. Nebel, "RT level makro modelling of leakage and delay under realistic PTV variation," in *Proc. LPonTR*, May 2008, pp. 31–32. [Online]. Available: https://www.staff.ncl.ac.uk/a.bystrov/LPonTR/2008/LPonTR-2008-Proceedings.pdf

[50] D. Helms, "Leakage models for high level power estimation," M.S. thesis, Deutsche Nationalbibliothek, Frankfurt, Germany, 2009, doi: 10.13140/RG.2.1.5166.7444.

[51] S. Narendra, S. Borkar, V. De, D. Antoniadis, and A. Chandrakasan, "Scaling of stack effect and its application for leakage reduction," in *Proc. ISLPED*, Huntington Beach, CA, USA, 2001, pp. 195–200.

[52] D. Helms *et al.*, Considering variation and aging in a full chip design methodology at system level," in *Proc. ESLsyn*, San Francisco, CA, USA, 2014, pp. 1–6.

[53] M. C. Hansen, H. Yalcin, and J. P. Hayes, "Unveiling the ISCAS-85 benchmarks: A case study in reverse engineering," *IEEE Des. Test Comput.*, vol. 16, no. 3, pp. 72–80, Aug. 2002.

Domenik Helms (M'13) received the graduation degree in theoretical physics and the Ph.D. degree in technical computer science from the University of Oldenburg, Oldenburg, Germany, in 2001 and 2009, respectively.

He is a Principle Scientist with OFFIS Institute, Oldenburg, where he has been the Manager of the Analysis of Nanometric ICs Group since 2009 and the Competence Center Embedded Systems Design Automation since 2011. His current research interests include analysis and optimization for non-functional aspects of recent transistor technologies, especially, leakage currents, process variation, and aging induced degradations.

Reef Eilers received the graduation degree in physics and the Ph.D. degree in technical computer science from the University of Oldenburg, Oldenburg, Germany, in 2009 and 2017, respectively.

He is a Senior Scientist with OFFIS Institute, Oldenburg. His current research interests include analysis and optimization for aging induced degradations, especially NBTI and HCI, and the influence of process variation on degradation.

Malte Metzdorf received the graduation degree in information technology from the Technical University of Kaiserslautern, Kaiserslautern, Germany, in 2010. He is currently pursuing the Ph.D. degree with the University of Oldenburg, Oldenburg, Germany.

He is a Senior Researcher with OFFIS Institute, Oldenburg, Germany. His current research interests include multiphysics simulations of digital systems covering thermal simulation, supply voltage simulation, power estimation, and aging estimation of a system on chip.

Wolfgang Nebel (SM'09–F'12) received the graduation degree in electrical engineering from the University of Hannover, Hanover, Germany, in 1982, and the Ph.D. degree in computer science from the Technical University of Kaiserslautern, Kaiserslautern, Germany, in 1986.

From 1987 to 1993, he was with Philips Semiconductor, Hamburg, Germany. Since 1993, he has been a Full Professor with the University of Oldenburg, Oldenburg, Germany. Since 1998, he has been a Board Member with OFFIS Institute, Oldenburg. He founded several IT start-up companies. He is the CEO of the OFFIS Institute, Oldenburg, and edacentrum, Hanover.

Prof. Nebel is the CEO of EDAA Society. He is the Vice President of the Konrad Zuse Society, and a member of the German National Academy of Science and Engineering (acatech).