Switch Level Time Simulation of CMOS Circuits with Adaptive Voltage and Frequency Scaling

Schneider, Eric; Wunderlich, Hans-Joachim

Proceedings of the IEEE VLSI TestSymposium (VTS'20), San Diego, US, 2020, pp. 1–6

doi: https://doi.org/10.1109/VTS48691.2020.9107642

Abstract: Design and test validation of systems with adaptive voltage-and frequency scaling (AVFS) requires timing simulation with accurate timing models under multiple operating points. Such models are usually located at logic level and compromise accuracy and simulation speed due to the runtime complexity.This paper presents the first massively parallel time simulator at switch level that uses parametric delay modeling for efficient timing-accurate validation of systems with AVFS. It provides full glitch-accurate switching activity information of designs under varying supply voltage and temperature. Offline statistical learning with regression analysis is employed to generate polynomials for dynamic delay modeling by approximation of the first-order electrical parameters of CMOS standard cells. With the parallelization on graphics processing units and simultaneous exploitation of multiple dimensions of parallelism the simulation throughput is maximized and scalable-design space exploration of AVFS-based systems is enabled. Results demonstrate the accuracy and efficiency with speedups of up to 159x over conventional logic level time simulation with static delays.

Preprint

Switch Level Time Simulation of CMOS Circuits with Adaptive Voltage and Frequency Scaling

Eric Schneider and Hans-Joachim Wunderlich
University of Stuttgart, Pfaffenwaldring 47, 70569 Stuttgart, Germany
schneiec@iti.uni-stuttgart.de, wu@informatik.uni-stuttgart.de

***Abstract*—Design and test validation of systems with adaptive voltage-and frequency scaling (AVFS) requires timing simulation with accurate timing models under multiple operating points. Such models are usually located at logic level and compromise accuracy and simulation speed due to the runtime complexity.**

This paper presents the first massively parallel time simulator at switch level that uses parametric delay modeling for efficient timing-accurate validation of systems with AVFS. It provides full glitch-accurate switching activity information of designs under varying supply voltage and temperature. Offline statistical learning with regression analysis is employed to generate polynomials for dynamic delay modeling by approximation of the first-order electrical parameters of CMOS standard cells. With the parallelization on graphics processing units and simultaneous exploitation of multiple dimensions of parallelism the simulation throughput is maximized and scalable-design space exploration of AVFS-based systems is enabled. Results demonstrate the accuracy and efficiency with speedups of up to 159× over conventional logic level time simulation with static delays.

***Keywords*— AVFS, parametric variations, switch level time simulation, GPU parallelization, statistical learning**

I. Introduction

In today's advanced nano-scaled technology systems parametrization and self-adaptation through *adaptive voltage and frequency scaling* (AVFS) is often employed to actively control internal supply voltages and clock frequencies of a system [1–3]. With AVFS, a system can trade-off power and performance to adapt to changing workloads, environmental conditions or performance degradation due to circuit aging [4]. This way, a system can overcome the *power wall* and run more reliably, which is of special interest in low-power-, embedded- as well as automotive applications [5, 6]. Since the performance has a high sensitivity towards process-, voltage- and temperature variations [7, 8], timing- and test-validation have become increasingly difficult. To enable meaningful design validation and design space exploration for AVFS-based systems, the designs needs to be thoroughly investigated under many different operating conditions [9, 10].

To validate the timing of AVFS-based systems, accurate simulation approaches with parametrizable dynamic delay modeling are required. Fig. 1 depicts the signal propagation in a small inverter cell under different supply voltages: Lower voltages lead to higher propagation delays at the output while higher voltages cause a speedup. Note that the slope of the signal is affected in addition which typically affects the signal propagation and timing at succeeding cells. Hence it is important to consider fine-grained voltage variations in the simulation models to provide reasonably accurate timing validation of current AVFS-based systems [10].

Several parametric delay models for voltage and temperature variations have been incorporated into gate level simulations. Existing approaches are typically based on either look-up tables, analytical models [13–16] or approximation techniques [7, 17]. However, conventional timing-accurate simulation is already very runtime-intensive and usually poses a bottleneck for larger designs with many input stimuli. By adding more complexity to the modeling, simulations and applications face severe scalability issues [18].

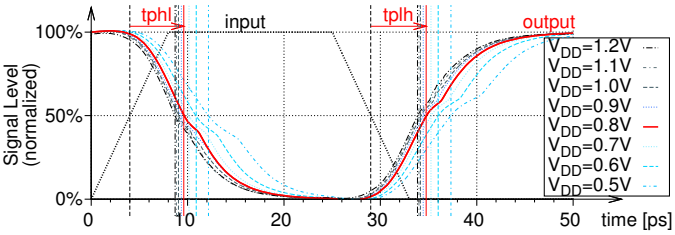

Fig. 1. Falling (tphl) and rising (tplh) transition propagation in a 15nm FinFET inverter cell [11, 12] under varying supply voltage in SPICE.

Besides parametric dependencies, transition ramps and pattern-dependent delays also impact circuit timing [19]. Yet, gate level simulation cannot cover all effects in a holistic and (at the same time) efficient manner. While simulations at electrical level (e.g., SPICE) provide a fine-grained modeling, they are not feasible for large-scale applications, due to high runtime and memory complexity. This even holds for their parallel implementations on massively-parallel compute architectures like graphics processing unit (GPU) accelerators [20, 21]. Thus, a modeling at intermediate abstraction levels is required, such as switch level [22], which recently regained interest also in cell-aware test [23]. Switch level timing simulation is able to consider transition ramps, pattern-dependent delays and glitch filtering *implicitly* in the modeling. By being more intuitive they allow for simpler, yet more detailed evaluations that can be effectively parallelized on GPUs [24].

This paper presents a novel switch level simulator with parametric delay modeling for scalable accurate timing validation of AVFS systems. First, cell characterization with statistical learning is applied to obtain functions that reflect the impact of voltage variations on the first-order electrical parameters in CMOS standard cells. During the actual simulation, the first-order electrical parameter functions are incorporated in the switch level modeling by approximation in a highly parallelized manner with maximal utilization of the computing throughput of the massively parallel GPUs. This way, scalable timing validation and design-space exploration of AVFS systems at switch level is enabled for the first time.

The remainder of the paper is organized as follows: The next section provides a brief background on parametric delay modeling. Section III then introduces the underlying switch level model. In section IV, the variation-aware switch level

parameter characterization pre-process is presented. Section V then explains the simulation flow and the GPU-parallelization. Finally, section VI demonstrates experimental results.

II. Background

Parametrizable delay models have been incorporated at gate level in various ways: Traditionally, look-up table-based models based on linear interpolation are utilized, which contain the propagation delay of each gate and every input pin for different process- and parameter corners [7]. To provide a reasonable accuracy, these tables have to be sufficiently large, but often their size grows exponentially with the number of parameters and they get even more complex when pattern-dependent delays have to be considered [25, 26].

Other models utilize closed-form expressions for analytical delay calculations [13–16]. For example, the α-power law model [13] states that the delay is proportional to the supply voltage V_{DD} simply through the relation

$$\tau \propto V_{\text{DD}}/(V_{\text{DD}} - V_{th})^{\alpha}, \tag{1}$$

where τ is the time constant, V_{th} is the transistor threshold voltage and $\alpha \in [1, 2]$ is a process-dependent parameter. In [14] an extension of the logical effort delay model [15, 27] is proposed, which is based on a simplified RC-modeling. It describes the propagation delay of a gate more generally by

$$d = \tau(gh + p), \tag{2}$$

with τ as a process-dependent delay constant, g as the *logical effort* which is gate-specific, h as the electrical effort (fanout) and p as additional parasitic impact. Parametric dependencies from process-, voltage- and temperature variations can be covered therein either by expressing a linear relationship in the logical effort, or by introducing so-called derrating coefficients deduced from non-linear dependencies [15]. Such analytical models typically assume independence between parameters for simplification. They also require a thorough understanding of the low-level impact and mechanisms, including additional information during evaluation such as signal slew rates [16].

Some delay models utilize approximation [7], where extensive SPICE transient analyses are run to extract gate delays under varying operating conditions. Multi-variable linear regression is then applied to find functions of fitting hypersurfaces that match the observed behavior, which are used to compute the gate delays during simulation. Similarly, in [17] an approximation technique was presented that utilizes neural networks to derive suitable functions for calculating the parametric gate delays. These approximations can be utilized to find fitting gate delays models for individual parameters and provide a flexible trade-off in accuracy and speed.

For current nanometer technology designs, the above methods are not sufficient to provide accurate simulations, since they apply to gate level only where many important CMOS-related timing effects are neglected. Instead, the paper at hand presents a novel parameter-variation-aware timing simulation at the more accurate switch level. The method comprises two distinct parts: 1) A pre-characterization phase done by the IC manufacturers and vendors in which the cell libraries are analyzed to generate variation-aware cell models considering first-order electrical parameters at switch level. 2) A massively parallel simulation phase in which the designers can efficiently evaluate their AVFS-designs with switch level accuracy based on the pre-generated first-order cell models.

III. First-Order Switch Level Modeling

The underlying switch level simulation model defines so-called Resistor-Resistor-Capacitor (RRC-) cells [24] as simulation entities, which are able to consider the major first-order electrical parameters found in CMOS standard cells [19] in an efficient and compact manner. RRC-cells are obtained from the transistor netlist of the circuit as shown in Fig. 2. First, the netlist is partitioned into *channel-connected components* (CCCs) [23, 28] each of which is mapped to an individual RRC-cell. For this, each transistor device D in the CCC is substituted by an input-controlled binary resistive switch $R^D := (V_{th}, R_{off}, R_{on})$, which can assume a conducting resistive state $R_{on} \in \mathbb{R}$ or a high-ohmic blocking state $R_{off} \in \mathbb{R}$. Given an input voltage $v \in \mathbb{R}$ at the transistor gate terminal, the R^D assumes a state depending on the transistor threshold voltage $V_{th} \in \mathbb{R}$.

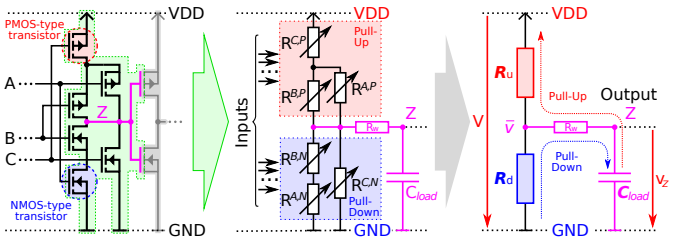

Fig. 2. Transistor netlist of a small circuit (left), extraction of an RRC-cell (center) with its functional abstraction (right) [24].

The transistors of pull-up and pull-down nets form a voltage-divider, composed of R_u and R_d, which charges (or discharges) an output load capacitance $C_{load} \in \mathbb{R}$. The voltage divider provides a stationary voltage $\overline{v} \in [V_{\text{GND}}, V_{\text{DD}}]$ between supply- (V_{DD}) and ground voltage (V_{GND}) and drives C_{load} via a wire resistance $R_w \in \mathbb{R}$ [24]. The voltage $v_Z(t) \in [V_{\text{GND}}, V_{\text{DD}}]$ at the output capacitor is time-dependent: If any transistor changes its state due to an input switch at some time $t_i \in \mathbb{R}$, $\overline{v}$ changes as well and C_{load} charges (or discharges) to the new stationary voltage via the effective resistance R_{eff} of the voltage divider and the wire R_w. This first-order transient response $v_Z(t)$ can be well described for $t \geqslant t_i$ after the input event by the following exponential equation:

$$v_Z(t) := (v_Z(t_i) - \overline{v}_i) \cdot e^{-\frac{\Delta t}{\tau_i}} + \overline{v}_i, \text{ with } t \geqslant t_i. \tag{3}$$

where $v_Z(t_i)$ is the output voltage at the beginning of the transient, $\tau_i := (R_{eff} + R_w) \cdot C_{load}$ is the time constant, $\overline{v}_i$ is the stationary voltage at the voltage divider and $\Delta t := (t - t_i)$ is the elapsed time since the transistor switch. In order to compensate for Miller-effects [19], the resulting output event is delayed by applying a small constant offset. The full switching history of a signal is modeled by piecewise approximation using exponential curves [24] which allows to efficiently represent time- and value-continuous voltage *waveforms*.

IV. Switch Level Parameter Characterization

In this work, an intuitive cell modeling based on polynomial approximation is employed that reflects variations

in supply voltage in all switch level transistor parameters $\rho \in \{V_{th}, R_{off}, R_{on}\}$. Before a simulation of a design can take place, the targeted standard cell libraries have to be characterized by the IC manufacturers to provide the corresponding polynomials for the variation-aware cell models. Besides the voltage, also variations in the ambient temperature are considered in this work, as both strongly contribute to the circuit delay. The ranges of the supply voltage $v \in [V_{min}, V_{max}]$ and the temperature $\theta \in [T_{min}, T_{max}]$ are assumed to be constrained by minimum and maximum values. Together they span a two-dimensional *parameter space* $\mathcal{P} \subset \mathbb{R}^2$ in which each point $P_i := (v_i, \theta_i) \in \mathcal{P}$ represents a distinct *operating point* for which the transistor parameter is assumed to have a corresponding value ρ_i. The *nominal* operating point is denoted as $P_{nom} \in \mathcal{P}$ and has the value ρ_{nom}.

A. Overview

The switch level device parameters and their parameter-dependencies are obtained in a characterization pre-process as shown in Fig. 3 which has to be performed only once for the used standard-cell library. The characterization flow starts with the extraction of unique transistor instances (i.e., FET-type, gate width and length, number of fins) used in the library cells (step A). Given a parameter type ρ of a transistor, the device is simulated in SPICE (step B) under a finite subset of operating points in $P_1, ..., P_m \in \mathcal{P}$ to obtain the corresponding transistor parameters values $\rho_1, ..., \rho_m$. The resulting data samples are then normalized, linearly interpolated and optionally further sub-sampled to obtain a denser data-grid (step C). Multi-variable linear regression is then applied to find a surface function that matches the data-grid (step D), which is then compiled for the use as transistor parameter functions (step E) during simulation. Here, the prior normalization of the data also avoids overfitting during regression.

In this work, each transistor parameter of the switch level model is calculated during the SPICE simulations based on the observation of the transistor I-V-characteristics. For this, the simple methods of [19] for obtaining the resistances R_{on}, R_{off} and threshold voltage V_{th} of a transistor are employed.

B. Parameter-variation-aware Switch Level Model

The impact of an operating point $P := (v, \theta)$ on a parameter ρ is expressed as deviation with respect to the nominal operating point $P_{nom} \in \mathcal{P}$ and its nominal value $\rho_{nom} \in \mathbb{R}$. For this, a surface function $f_\rho : \mathcal{P} \to \mathbb{R}$ is constructed that approximates the deviation of the selected transistor parameter type ρ for different $P_i \in \mathcal{P}$ with small error such that

$$\forall P_i \in \mathcal{P} : f_\rho(P_i) \approx \frac{\rho_i}{\rho_{nom}} - 1. \tag{4}$$

For the calculation of the transistor parameters, higher-order multi-variable polynomials are used which are able to approximate continuous differentiable hypersurfaces within constrained intervals. The degree of accuracy of the approximation typically increases with the order N of each variable. The generalized form of a two-dimensional polynomial of order $2 \cdot N$ is defined as $f_\rho : \mathcal{P} \to \mathbb{R}$,

$$f_\rho(P) := \sum_{i=0}^{N} \sum_{j=0}^{N} \left(\beta_{i,j} \cdot v^i \theta^j\right), \text{ with } P := (v, \theta) \in \mathcal{P}, \tag{5}$$

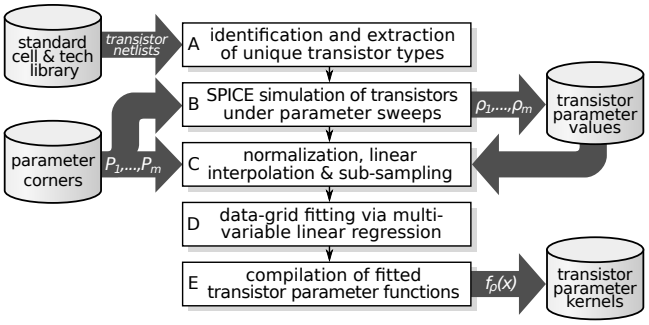

Fig. 3. Device parameter characterization and kernel generation pre-process.

where each product term consists of a coefficient $\beta_{i,j} \in \mathbb{R}$ for the corresponding power term of the predictor variables $v^i \theta^j$.

C. Defining Functions by Multi-variable Linear Regression

The coefficients $\beta_{i,j}$ of the surface function are *unknown* beforehand and need to be determined. For this, multi-variable linear regression is used, which allows to quickly find coefficients of a fitting surface polynomial. In multi-variable linear regression, an equation system is set up based on the parameter values obtained from SPICE simulation of a transistors type under different operating points. Given a set of $m \in \mathbb{N}$ data samples $S := \{P_i \in \mathcal{P} | i = 1, ..., m\}$, the linear regression model can be represented in matrix-form as

$$y = \mathbf{X}\beta + \varepsilon \tag{6}$$

where

$$y = \begin{pmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_m \end{pmatrix}, \mathbf{X} = \begin{pmatrix} v_1^0\theta_1^0 & v_1^0\theta_1^1 & v_1^1\theta_1^0 & \dots & v_1^N\theta_1^N \\ v_2^0\theta_2^0 & v_2^0\theta_2^1 & v_2^1\theta_2^0 & \dots & v_2^N\theta_2^N \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v_m^0\theta_m^0 & v_m^0\theta_m^1 & v_m^1\theta_m^0 & \dots & v_m^N\theta_m^N \end{pmatrix}, \beta = \begin{pmatrix} \beta_{0,0} \\ \beta_{0,1} \\ \beta_{1,0} \\ \vdots \\ \beta_{N,N} \end{pmatrix} \tag{7}$$

with residual $\varepsilon \in \mathbb{R}^m$. The entries of a row k in the matrix $\mathbf{X} \in \mathbb{R}^{m \times (N+1)(N+1)}$ correspond to the polynomial terms $v_k^i \theta_k^j$ for sample $P_k \in S$. The entries in a column l correspond to the l-th order polynomial terms. The first column values typically reflect the zero-degree power terms which evaluate to 1.

All operating point parameters and corresponding transistor values in the above equation system are normalized prior to regression to evenly weight the parameters and to avoid overfitting. For the operating points $P := (v, \theta)$, the normalization function $\phi_\mathcal{P} : \mathcal{P} \to [0, 1]^2, \phi_\mathcal{P}(P) := (\frac{v - V_{min}}{V_{max} - V_{min}}, \frac{\theta - T_{min}}{T_{max} - T_{min}})$ is used. The transistor parameters ρ are normalized using the function $\phi_D : \mathbb{R} \to \mathbb{R}, \phi_D(\rho) := \frac{\rho}{\rho_{nom}} - 1$, which describes the parameter value deviation with respect to the nominal operating point P_{nom}.

A viable solution β for the system in Eq. (6) is obtained by following the *ordinary least squares* criterion [29]. For this, coefficients $\hat{\beta} \in \mathbb{R}$ are fitted through minimization of the squared residuals in the Euclidean L^2-norm $|| \cdot ||_2$:

$$\hat{\beta} = \arg\min_{\beta} \{||y - \mathbf{X}\beta||_2^2\}. \tag{8}$$

This will eventually deliver suitable coefficients $\hat{\beta}_{i,j} \in \mathbb{R}$ that allow to evaluate the polynomial $f_\rho(P)$ in Eq. (5) and determine the deviation of the transistor parameter ρ from ρ_{nom}. Each polynomial is then solely identified by its

coefficient vector. Note that the polynomial approximation is very sensitive to deviations in the coefficients. Therefore all operations during regression as well as the evaluation of the polynomial require double-precision floating point operations.

V. High-Throughput Parameter-aware Simulation

The switch level parameter polynomials generated for the cell library can be used by designers to efficiently simulate AVFS circuits under voltage and temperature variations in parallel on GPUs. For this, all coefficient vectors are stored in an constant double-precision floating point array in the global GPU device memory. Transistor type and parameter index are used to address the corresponding coefficient vectors for evaluation. The evaluation of the polynomial is implemented as kernel function that can be accessed by any thread on the GPU during waveform evaluation. Each call to the kernel accepts an individual coefficient vector $\beta \in \mathbb{R}^{(N+1)\cdot(N+1)}$ and normalized operating point $P \in \mathcal{P}$ as arguments for which the corresponding parameter function is evaluated by the thread. Hence, although the evaluation of the polynomial by different threads involves the exact same function calls for each transistor parameter, the selected coefficients determine the actual computed transistor function.

A. Evaluation of Transistor Parameter Functions

In general, each thread on the GPU is responsible for the processing of a single RRC-cell in the circuit for a provided waveform stimuli and operating point. For the parameter-variation-aware waveform processing a thread has to:

1) load the cell description with nominal transistor parameters D from global to thread-local private memory,
2) read assigned operating point parameters P,
3) select transistor parameter $\rho_{nom} \in D$ of local description,
4) fetch coefficients β of corresponding transistor parameter function and calculate parameter deviation $f_\rho(P)$
5) adapt parameter in locally stored transistor description D:

$$\rho' := \rho_{nom} \cdot (1 + f_\rho(P)) \tag{9}$$

The cell evaluation thread repeats step (3) to (5) for every single transistor parameter of each transistor in the RRC-cell description. Afterwards it continues with the main waveform processing loop [24] to compute the corresponding output.

B. Parallelization

The key factor of achieving high simulation speedup is achieved by aiming for high simulation throughput from the exploitation of multiple dimensions of parallelism. Basically, all parallel threads of the simulation and evaluation kernels are organized in multi-dimensional arrays [30] as shown in Fig. 4, which simultaneously exploit available parallelism from both structure and data. Threads in the vertical dimension form a simulation *slot*, in which the data-independent RRC-cells of a topological level in a circuit instance are processed concurrently (*node-parallelism*). In the horizontal dimension, the threads form two-dimensional planes, in each of which a thread evaluates a certain node of the current level to be processed for different input stimuli waveforms (*waveform-parallelism*) as well as operating points applied to the respective circuit instance (*instance-parallelism*). Note that, despite

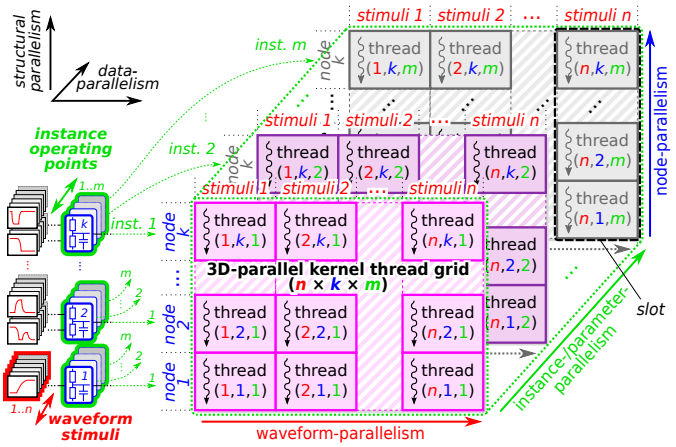

Fig. 4. Multi-dimensional parallel thread-grid organization for massively parallel fault simulation with n stimuli waveforms under m operating points.

individual assignments of waveforms or parameters to the threads, they compute the same instructions in a data-parallel fashion. Hence, all of the threads follow the SIMD-execution scheme without additional control flow divergence, since the node functions and kernel calls remain the same, thereby achieving maximum simulation throughput.

The parallelization scheme allows to arbitrarily trade-off between the simulation of multiple stimuli and multiple operating points at any time. This flexibility is prerequisite to maximize the slot utilization on the GPU and therefore to achieve highest simulation throughput. With the high arithmetic computing capabilities of the GPUs, the increased complexity of the delay calculations can be handled well and efficiently, even though many additional floating point multiplications and additions need to be performed in double-precision.

VI. Experimental Results

In the following the NanGate 15nm Open Cell Library [12] is investigated for modeling, characterization and simulation. For the characterization of the transistor parameters (V_{th}, R_{on} and R_{off}), discrete operating point samples $S := \{(V_{\mathrm{DD}i}, \theta_j) \mid i,j = 1,2,...\}$ were chosen with the supply voltage $V_{\mathrm{DD}i} \in [0.5\mathrm{V}, 1.2\mathrm{V}]$ in steps of 0.05V (nominal 0.8V) and ambient temperatures $\theta_j \in [-25^\circ\mathrm{C}, 125^\circ\mathrm{C}]$ in steps of 12.5°C (nominal 25°C). A commercial SPICE simulator was used to run the parameter sweeps which took only a few seconds for each transistor parameter. The regression analysis was implemented in Python and took much less than a second. Again, the above steps are a pre-process and need to be performed once only for each transistor type. To evaluate the simulation, the largest designs from ISCAS'89, ITC'99 and industrial benchmarks were synthesized in a commercial synthesis tool flow. During the process all sequential elements were removed such that only the combinational logic remained (full-scan). Transition delay test patterns were generated for each design using a commercial ATPG-tool which were topped-off with timing-aware patterns targeting the 200 longest paths reported from a timing-analysis tool. All experiments were run on a host system consisting of two Intel®Xeon E5-2687W v2 processors clocked at 3.4GHz with 256GB of main memory and a NVIDIA® Tesla™ P100 GPU (CUDA 9.2) with 3584 cores and 16GB global device memory.

A. Regression Analysis

Fig. 5 depicts the distribution of the transistor parameter approximation errors of the two-dimensional polynomials for orders $N := 1, ..., 7$ in each variable. The error was computed for a grid of 100×100 equidistant operating points in the parameter space, whose values were compared to the surface of the piecewise linear interpolation of the sample grid spanned by S. On the right, a magnification of the error range between -2% and 2% is shown for the higher-order polynomials.

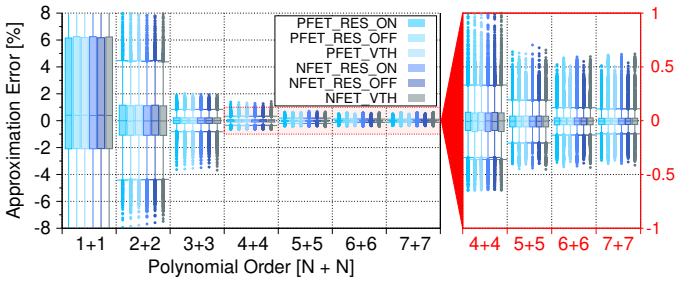

Fig. 5. Approximation error distribution of the transistor parameter polynomials in 100×100 equidistant operating points in [0.5V,1.2V] × [-25°C,125°C].

The approximation error decreases for higher order polynomials and the mean and standard deviation of the error distributions quickly converge to 0.0001% and 0.11%, respectively. For polynomials of order "3+3" ($N = 3$ for each predictor) the maximum error is already below 4% and eventually falls below 1% for orders "5+5" and higher. Note that while each tile of the interpolated data set itself is *flat*, the complete surface is not continuously differentiable. Thus, the approximation error of the polynomial surface shows spikes when crossing the boundaries of different tiles (cf. Fig. 6-b).

Fig. 6-a) illustrates the result of the "5+5" order polynomial approximation of the conducting resistance (R_{on}) of a PFET-type transistor. Black points represent sample operating points that have been explicitly evaluated in SPICE, while the shaded surface corresponds to the linear interpolation. As indicated by the overlap of the contour lines, the polynomial approximation fits the reference surface well. This claim is also supported by Fig. 6-b) which shows that the relative approximation error is within 1%. Again, the spikes in the error surface (cf. "const. 25°C") result from the curved polynomial approximation passing tiles of the linear interpolation.

B. Simulation Performance

Table I summarizes the simulation performance and speedup for each design. In column 2 and 3 the size of the circuit in number of nodes (cells + circuit ports) and the size of the transition delay test set in test pairs is shown. Column 4 and 5 contain the runtime for evaluating the pattern set as well as the throughput performance in *million (node) evaluations per second* (MEPS) of a serial commercial *logic level* time simulator with static timing annotations. Also, the simulation time of the baseline GPU-accelerated switch level simulation from [24] is shown in column 6. Finally, the last three columns present the runtime, the throughput performance and speedup of the presented parameter-aware switch level simulation approach over the commercial logic level solution (cf. Col. 4). The polynomial order of the approximation has been chosen as "5+5" ($N = 5$ for each predictor).

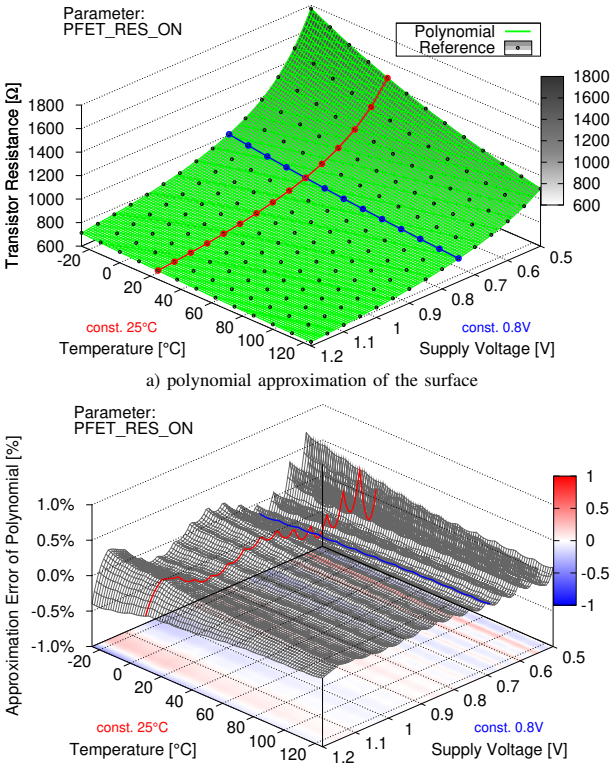

Fig. 6. Approximation of the PFET ON-resistance (conducting state R_{on}) as obtained from SPICE using a surface polynomial of order $2 \cdot N$ with $N = 5$.

TABLE I
CIRCUIT STATISTICS AND RUNTIME PERFORMANCE (V_{DD}=0.8V AT 25°C).

Circuit[1]	Nodes[2]	Test Pairs[3]	Event-Driven Time[4]	Event-Driven MEPS[5]	[24] Time[6]	Proposed (order 5+5) Time[7]	Proposed (order 5+5) MEPS[8]	Proposed (order 5+5) X[9]
s38417	18999	173	2.25s	1.46	30ms	29ms	112.6	78
s38584	23053	194	2.71s	1.65	38ms	39ms	112.7	69
b17	42779	818	16.71s	2.09	289ms	294ms	118.7	57
b18	125305	961	1:52m	1.07	1.22s	1.24s	97.1	91
b19	250232	1916	7:31m	1.06	4.50s	4.58s	104.7	99
b20	18384	760	13.36s	1.05	169ms	172ms	81.0	78
b21	19253	749	11.45s	1.26	177ms	179ms	80.2	64
b22	27847	692	16.12s	1.20	227ms	228ms	84.3	71
p35k	47997	3298	1:14m	2.12	1.24s	1.42s	111.5	53
p45k	44098	2320	45.27s	2.26	893ms	904ms	113.2	51
p100k	96172	2211	3:11m	1.11	1.84s	1.85s	115.1	104
p141k	178063	995	2:22m	1.24	1.39s	1.37s	129.0	105
p418k	440277	1516	8:07m	1.37	4.43s	4.65s	143.6	105
p500k	527006	3820	0:34h	0.96	15.53s	16.04s	125.5	131
p533k	676611	1940	0:30h	0.71	11.72s	11.64s	112.8	159
p951k	1090419	4080	1:11h	1.04	28.01s	28.27s	157.4	151
p1522k	1088421	8021	2:06h	1.15	1:04m	1:05m	134.1	117

As shown, the runtimes for the serial simulation of the test pattern set in the commercial logic level simulator ranges from a few seconds for the smaller and over two hours for the largest design. Thus, in average a throughput performance of 1.3 MEPS was obtained. The presented simulation approach ran the simulations with up to 157 MEPS and an average speedup of 93× despite the more complex simulation model. While the simulation of the largest circuit took roughly a minute, the highest speedup of 159× observed was obtained for the circuit p533k. In general, the speedup is higher for the larger designs due to the larger amount of available parallelism. When comparing the runtimes to the baseline algorithm with static parameters [24] (cf. Col. 6), no significant overhead from the additional calculations of the polynomials was observed. This holds even for the higher order polynomials, as

the overall simulation runtime is dominated by the switch level waveform evaluation and the memory overhead. The setup time of the presented simulator typically required roughly up to 90 seconds (for p1522k), without costly code-compilation and optimization of the netlists and the timing annotations. Therefore, only bare simulation times were considered to allow for an unbiased comparison.

C. Modeling Accuracy

In the following, the presented simulation model is validated for an example circuit composed of a chain of 20 inverter cells being simulated under ten different operating conditions. Fig. 7 compares the waveforms obtained at stage 7 and 20 from SPICE and switch level simulation. The color and the stroke type of the waveforms indicate the applied supply voltage and temperature, respectively. The amplitude of each waveform is normalized according to the applied supply voltage.

As shown, the impact of voltage and temperature are well reflected in the switch level model: Higher supply voltages lead to a significantly faster circuit, while lower voltages cause a slowdown. Also, higher temperatures speed up the circuit, which is a phenomena commonly observed in FinFET technology [31]. The resulting switch level waveforms are able to reflect many characteristics found in CMOS technology. Thus, with the presented parametric switch level simulation, SPICE-like accuracy can be achieved with runtimes even faster than conventional simulation at logic level with static timing.

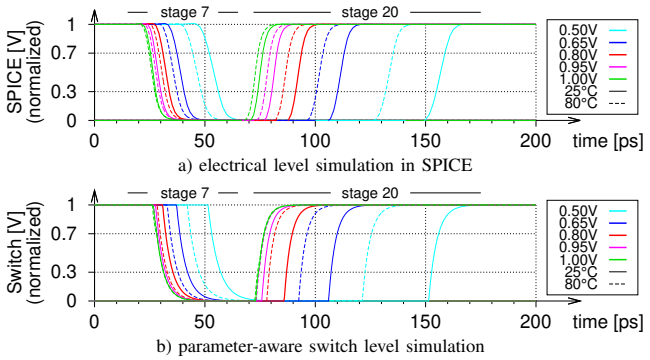

Fig. 7. Signal propagation under varying operating conditions in a chain of 20 inverter cells [12] with signals shown at stage 7 and 20. The line colors and stroke types indicate the applied V_{DD} and temperature, respectively.

VII. Conclusion

This work presented an approach for parameter-aware switch level time simulation of AVFS-based systems for the execution on massively parallel GPUs. The basic switch level model covers all major first order electrical parameters found in CMOS cells and approximates the supply voltage- and temperature-dependent behavior. The delay modeling utilizes statistical learning to capture the dynamic behavior and parametric dependencies, and is able to compute delays with negligible overhead. Exploitation of multiple dimensions of parallelism, including the evaluation of many operating points in parallel, allow to maximize the simulation throughput for fast and large-scale design validation and exploration of AVFS-based systems. Compared to a commercial serial logic level time simulation with static delays, the presented switch level approach was able to achieve speedups of over 150× despite its more detailed and accurate modeling.

Acknowledgment

This work is part of the project grant WU 245/19-1 funded by the German Research Foundation (DFG).

References

[1] T. Kuroda, "CMOS Design Challenges to Power Wall," in *Proc. Int'l Microprocesses and Nanotechnology Conf. Digest of Papers.*, Oct. 2001, pp. 6–7.
[2] M. Horowitz, E. Alon, D. Patil *et al.*, "Scaling, Power, and the Future of CMOS," in *Proc. IEEE Int'l Electron Devices Meeting (IEDM)*, Dec. 2005, pp. 7 pp.–15.
[3] S. Borkar and A. A. Chien, "The Future of Microprocessors," *Communications of the ACM*, vol. 54, no. 5, pp. 67–77, May 2011.
[4] J. Tschanz, N. S. Kim, S. Dighe *et al.*, "Adaptive Frequency and Biasing Techniques for Tolerance to Dynamic Temperature-Voltage Variations and Aging," in *Proc. IEEE Int'l Solid-State Circuits Conf. (ISSCC)*, Feb. 2007, pp. 292–604.
[5] S. Kiamehr, M. Ebrahimi, and M. Tahoori, "Temperature-aware Dynamic Voltage Scaling for Near-Threshold Computing," in *Proc. Int'l Great Lakes Symp. on VLSI (GLSVLSI)*, May 2016, pp. 361–364.
[6] S. Mhira, V. Huard, A. Benhassain *et al.*, "Dynamic Adaptive Voltage Scaling in Automotive environment," in *Proc. IEEE Int'l Reliability Physics Symp. (IRPS)*, Apr. 2017, pp. 3A–4.1–3A–4.7.
[7] A. Srivastava, D. Sylvester, and D. Blaauw, *Statistical Analysis and Optimization for VLSI: Timing and Power*, 1st ed. Springer, 2005.
[8] E. Amat, A. Calomarde, and A. Rubio, "Reliability Study on Technology Trends Beyond 20nm," in *Proc. 20th Int'l Conf. on Mixed Design of Integrated Circuits and Systems (MIXDES)*, Jun. 2013, pp. 414–418.
[9] S. Borkar, T. Karnik, S. Narendra *et al.*, "Parameter Variations and Impact on Circuits and Microarchitecture," in *Proc. Design Automation Conf. (DAC)*, June 2003, pp. 338–342.
[10] I. Polian, B. Becker, S. Hellebrand, H. Wunderlich, and P. Maxwell, "Towards Variation-Aware Test Methods," in *Proc. IEEE 16th European Test Symp. (ETS)*, May 2011, pp. 219–225.
[11] K. Bhanushali and W. R. Davis, "FreePDK15: An Open-Source Predictive Process Design Kit for 15Nm FinFET Technology," in *Proc. Int'l Symp. on Physical Design (ISPD)*, Mar. 2015, pp. 165–170.
[12] NanGate Inc., "NanGate15nm Open Cell Library." http://www.nangate.com/, 2017.
[13] T. Sakurai and A. R. Newton, "Alpha-Power Law MOSFET Model and its Applications to CMOS Inverter Delay and Other Formulas," *IEEE Journal of Solid-State Circuits*, vol. 25, no. 2, pp. 584–594, Apr. 1990.
[14] C.-H. Wu, S.-H. Lin, and H. Chiueh, "Logical Effort Model Extension with Temperature and Voltage Variations," in *Proc. 14th Int'l Workshop on Thermal Inveatigation of ICs and Systems*, Sept 2008, pp. 85–88.
[15] B. Lasbouygues, R. Wilson, N. Azemard, and P. Maurine, "Temperature and voltage aware timing analysis: Application to voltage drops," in *Proc. Conf. on Design, Automation Test in Europe (DATE)*, Apr. 2007, pp. 1–6.
[16] K. Shinkai, M. Hashimoto, and T. Onoye, "A gate-delay model focusing on current fluctuation over wide range of process-voltage-temperature variations," *Integration, the VLSI Journal*, vol. 46, no. 4, pp. 345–358, 2013.
[17] B. P. Das, V. Janakiraman, B. Amrutur, H. S. Jamadagni, and N. V. Arvind, "Voltage and Temperature Scalable Gate Delay and Slew Models Including Intra-Gate Variations," in *Proc. 21st Int'l Conf. on VLSI Design (VLSID)*, Jan. 2008, pp. 685–691.
[18] J. Mahmod, S. Millican, U. Guin, and V. Agrawal, "Special Session: Delay Fault Testing - Present and Future," in *Proc. IEEE 37th VLSI Test Symp. (VTS)*, Apr. 2019, pp. 1–10.
[19] N. H. E. Weste and D. M. Harris, *CMOS VLSI Design – A Circuits and Systems Perspective*, 4th ed. Addison-Wesley, 2011.
[20] K. Gulati, J. F. Croix, S. P. Khatri, and R. Shastry, "Fast Circuit Simulation on Graphics Processing Units," in *Proc. 14th Asia and South Pacific Design Automation Conf. (ASP-DAC)*, Jan. 2009, pp. 403–408.
[21] L. Han, X. Zhao, and Z. Feng, "TinySPICE: A Parallel SPICE Simulator on GPU for Massively Repeated Small Circuit Simulations," in *Proc. ACM/EDAC/IEEE 50th Design Automation Conf. (DAC)*, May 2013, pp. 1–8, Article 89.
[22] R. E. Bryant, "A Switch-Level Model and Simulator for MOS Digital Systems," *IEEE Transactions on Computers (TC)*, vol. C–33, no. 2, pp. 160–177, Feb. 1984.
[23] H. H. Chen, S. Y.-H. Chen, P.-Y. Chuang, and C.-W. Wu, "Efficient Cell-Aware Fault Modeling by Switch-Level Test Generation," in *Proc. IEEE 25th Asian Test Symp. (ATS)*, Nov. 2016, pp. 197–202.
[24] E. Schneider, S. Holst, X. Wen, and H.-J. Wunderlich, "Data-Parallel Simulation for Fast and Accurate Timing Validation of CMOS Circuits," in *Proc. IEEE/ACM 33rd Int'l Conf. on Computer-Aided Design (ICCAD)*, Nov. 2014, pp. 17–23.
[25] V. Chandramouli and K. A. Sakallah, "Modeling the Effects of Temporal Proximity of Input Transitions on Gate Propagation Delay and Transition Time," in *Proc. 33rd Design Automation Conf. (DAC)*, Jun. 1996, pp. 617–622, Paper 40.2.
[26] L.-C. Chen, S. K. Gupta, and M. A. Breuer, "A New Gate Delay Model for Simultaneous Switching and Its Applications," in *Proc. 38th Design Automation Conf. (DAC)*, Jun. 2001, pp. 289–294, Paper 19.2.
[27] I. Sutherland, B. Sproull, and D. Harris, *Logical Effort: Designing Fast CMOS Circuits*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999.
[28] A. E. Ruehli and G. S. Ditlow, "Circuit Analysis, Logic Simulation, and Design Verification for VLSI," *Proceedings of the IEEE*, vol. 71, no. 1, pp. 34–48, Jan. 1983.
[29] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. Springer New York, 2009.
[30] E. Schneider and H.-J. Wunderlich, "SWIFT: Switch Level Fault Simulation on GPUs," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Syst. (TCAD)*, vol. 38, no. 1, pp. 122–135, Jan. 2019.
[31] S. Soleimani, A. Afzali-Kusha, and B. Forouzandeh, "Temperature Dependence of Propagation Delay Characteristic in FinFET Circuits," in *Proc. Int'l Conf. on Microelectronics (ICM)*, Dec. 2008, pp. 276–279.