Accurate Estimation of CMOS Power Consumption Considering Glitches by Using Waveform Lookup

Michael Meixner and Tobias G. Noll

***Abstract*—Gate-level power estimation methodologies are often considered as a sign-off level reference for digital circuit design. Nevertheless, when gate delays and related effects like glitches are taken into account, commercial state-of-the-art gate-level power estimators show surprisingly large estimation errors. Following an analysis of factors causing these inaccuracies, a novel gate-level power estimation approach is proposed, which combines lookup-based macromodels with the accuracy of analog signal waveforms and achieves significantly better results under the influence of glitches.**

***Index Terms*—Dynamic energy dissipation, gate delay effects, gate level, glitches, power estimation.**

I. Introduction

PHYSICAL circuit simulation is the most reliable source for power estimation prior to measurement of silicon. However, due to the complexity of today's circuits, sign-off level power estimations are commonly performed on higher levels of abstraction particularly on the gate level where established estimation flows exist. Power analysis on the gate level abstracts from currents and voltages by only considering switching events and boolean logic states. The related power consumption is derived from characterized lookup tables. The abstraction from numerous physical effects that allows for speedup leads to inaccuracies. Benchmarks of gate-level power estimators which are detailed in the last section revealed errors of up to 39% and a mean error of 13% for the estimated dynamic energy dissipation compared to circuit simulation.

This work aims at improving power estimation accuracy on the gate level. The focus is on effects due to signal delays that are handled poorly by current estimation approaches but often significantly contribute to power consumption. Despite the rising influence of leakage power, for most applications the main contribution to power consumption is still caused by switching of circuit nodes. An accurate prediction of switching activities therefore promises high estimation accuracy. Due to glitches which are caused by unaligned transitions on different inputs of gates the outputs might feature multiple transitions in a single clock cycle. These spurious transitions typically form a contribution of 20%–30% of the power consumption but depending on circuit topology can reach contributions of up to 70% [1] not least because once a glitch is created it might propagate through downstream gates. Accurately predicting the number and properties of these glitches on higher levels of abstraction is challenging, as they are strongly dependent on physical circuit properties.

Early efforts to consider glitching in power estimation focused on evaluation of switching activity by modified simulators (e.g., [2] and [3]) or probabilistic means (e.g., [4] and [5]). When they were considered at all, partial-swing transitions were determined in a postprocessing step to derive scaling factors to the power consumption of the related full-swing transitions. In addition, there have been several approaches of encapsulation of functional blocks inside macromodels that consider glitches on the inputs. Examples include models for components of digital arithmetic such as adders or multipliers [6], [7] or even full subcircuits [8]. These approaches require the estimated circuit to be composed of the characterized word-level components. Depending on model size, this severely restricts general applicability. In addition, due to the fundamental changes to the estimation flow these high-level models are generally more suitable for design space exploration than for the analysis of actual designs. In contrast, the proposed approach aims at improving power estimation by accurate handling of glitches without sacrificing flexibility or requiring adoption of different design styles.

II. Shortcomings of State-of-the-Art Gate-Level Power Estimators

In the first phase of state-of-the-art gate-level power estimation, the switching activities of all nodes in the netlist are determined. These activities are used in the second phase to look up the related power consumption from the cell library. To be able to consider delay effects like glitches, the evaluation of switching activity needs to take realistic gate delays into account. These load-dependent delays can be estimated by static timing analysis and annotated for logic simulation. As an alternative, probabilistic methods can be used to estimate switching activities. These approaches suffer from similar problems as logic simulation but introduce additional challenges like handling of signal correlation due to reconvergent fanout. The shortcomings of state-of-the-art power estimation highlighted in the following paragraphs will serve as a motivation for the proposed estimation approach.

A. Errors Due to Logic Simulation

A limitation that is inherent to logic simulation is the restriction to two signal levels and the abstraction from finite transition slopes. In contrast, glitches occurring during actual operation of the circuit do not necessarily feature full-swing

Manuscript received August 25, 2016; revised October 23, 2016; accepted December 17, 2016. Date of publication December 20, 2016; date of current version June 23, 2017. This brief was recommended by Associate Editor D. Chen.

The authors are with the Chair of Electrical Engineering and Computer Systems, RWTH Aachen University, 52056 Aachen, Germany (e-mail: meixner@eecs.rwth-aachen.de; tgn@eecs.rwth-aachen.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSII.2016.2642179

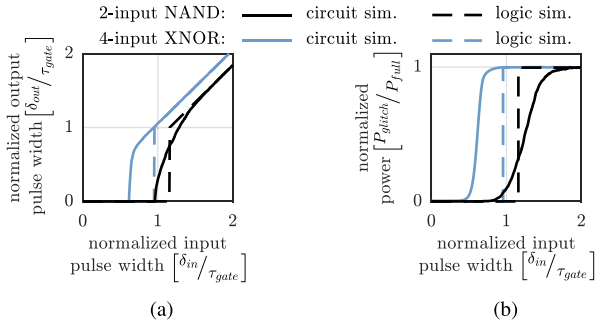

Fig. 1. Example of glitch propagation and related switching power. (a) Output pulse width. (b) Power consumption.

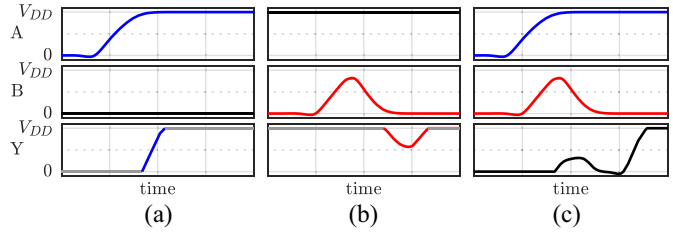

Fig. 2. Output response of a two-input XOR gate to different input events. (a) Transition on A. (b) Pulse on B. (c) Both input events.

transitions due to limited signal slopes. The second effect that is hard to model in logic simulation is the impact of a glitch on the input of a gate on the output. In logic simulators this propagation characteristic is commonly implemented by using inertial gate delays [9] which prevent pulses that are shorter than the gate delay from propagating to the gate output. This is a simplification since the physical effect of a short pulse at the input is often visible as a partial swing pulse at the output. Fig. 1(a) shows the relationship between the input and output pulse widths for a NAND gate and a multistage XNOR gate. As opposed to logic simulation with glitch filtering, circuit simulation does not show an abrupt cut-off of the propagation of short input pulses. Fig. 1(b) demonstrates that pulses that do not feature full swing transitions may nevertheless significantly contribute to power consumption. Inertial glitch filtering causes overestimated power consumption when small glitches are applied to the NAND gate while it would potentially lead to underestimation of the XNOR gate by rejecting too many glitches. This simple example demonstrates that the gate delay alone is not a good measure for glitch filtering.

B. Errors Due to Energy Lookup

Additional shortcomings are caused by simplifications of the cell libraries, most fundamentally by the restriction to the characterization of single-input events [10]. Therefore, power consumption related to quasisimultaneous switching of multiple inputs can only be approximated. This problem is most severe for gates like full adders or XOR gates for which switching of the individual inputs always results in switching of the output while quasisimultaneous switching of multiple inputs might not cause the output state to change. As the lookup tables in the cell library for these cells do not contain events for switching of inputs without activity on the output, no matching power values can be found for either of the switching events and the power cannot be estimated at all.

III. Possible Modifications of the Estimation Methodology

The restriction to single-input events could be avoided by the adoption of a more complex characterization that includes multi-input events as well. Because of the critical importance of the timing offset between switching on multiple inputs, a significant number of offsets would have to be considered during characterization. This change would be costly in terms of characterization effort but could potentially reduce the estimation errors with only a small overhead in estimation runtime. In contrast, the simplified handling of glitches cannot be fully corrected by preprocessing or postprocessing of data relevant to the logic simulator. A proposed method targeting this shortcoming appends information regarding the transition slopes to each signal change [11]. Using a custom characterization of the gates, this allows for more accurate filtering of glitch pulses. Pulses that are prevented from propagation during simulation are used in a postprocessing step to consider the energy due to partial-swing pulses. This approach cannot adequately handle correlations between multiple transitions on different inputs especially when one of the inputs features a partial swing glitch pulse. This problem is demonstrated in Fig. 2, which shows the inputs A and B and output Y of an XOR gate. The first two columns show the response to a full transition and a short pulse on one of the inputs, respectively. The third column demonstrates that the response to quasisimultaneous stimulation with both events significantly differs from the superposition of the individual responses.

IV. Event-Based Lookup of Power Waveforms

Handling of complex delay effects leading to generation and filtering of glitches requires more information on the switching waveforms than produced by logic simulation. The proposed approach combines the runtime advantages of lookup-based power estimation with the accuracy of analog signal waveforms. It extends the lookup tables by multi-input events with varying offsets as well as events involving glitches. In addition, the analog waveform response at the output is stored in the library for each event combination along with the supply current waveform which defines the power consumption. This output waveform is used as an accurate input during power estimation of the next gate, thereby allowing consecutive processing of the whole circuit. This completely removes logic simulation from the estimation flow, and therefore, avoids all error sources related to suboptimal handling of glitches during simulation. Estimation for each gate is performed in multiple steps.

1) The signal characteristics of the nets connected to the inputs are retrieved from the local database.
2) Switching events on different inputs that occur in close temporal proximity are grouped into event combinations.
3) The supply current as well as the output voltage response of each combination is looked up from the library.
4) The potentially overlapping voltage and current responses are combined. The resulting supply current waveform is used to calculate the power consumption.

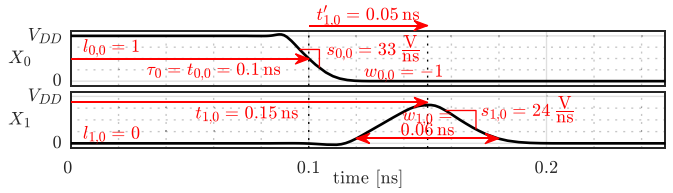

Fig. 3. Annotated characteristics for an event combination K_0 consisting of two switching events.

5) The voltage waveform at the output is analyzed to retrieve the relevant signal characteristics which are stored in the local database for use in subsequent estimations.

A. Signal Characteristics for Power Waveform lookup

The method discussed here is similar to logic simulators in that individual *events* on circuit nodes are considered. However, in contrast to logic simulation, an event is not simply defined by a change of the logic state at a time t but is annotated with more detailed characteristics. One of these properties is the transition slope s of each event. For glitch propagation, a vital property is the shape of the pulse which is characterized by its width w. This way partial- and full-swing glitches can be equally described and the pulse height can be determined from the transition slope. A full-swing transition is denoted by a pulse width $w = -1$. An example for this annotation of characteristics to each event is depicted in Fig. 3 for both a full-swing transition and a partial-swing pulse. Event times are defined at 50% signal swing or maximum of the pulse, respectively, pulse widths are measured at 10% of signal swing and slopes are considered from 10% to 90% thresholds.

Each analyzed signal is decomposed into switching events that correspond to those considered during characterization. A crucial parameter is the threshold that determines whether two consecutive full-swing transitions are considered as a glitch or as two separate events. Especially, for complex multistage gates and multi-input events the characterization of full-swing glitches compared to the concatenation of two separate transitions can result in a large gain in accuracy due to increasing importance of interactions. A suitable compromise between characterization effort and accuracy was found to be an upper glitch width of the maximum of either the gate delay or four times the transition time.

B. Grouping Into Event Combinations

Since the influence of each input event on the output and supply current is limited in time to a gate-specific interval δ_t, interdependencies between events on multiple inputs only need to be considered for events within a maximum distance of δ_t from each other. Each event combination K_j is defined as the ordered list of events on all N gate inputs. To derive a unique representation of the event combination that corresponds to the library, all time offsets t'_i are defined relative to the earliest event in the combination j which is retained as the combination time offset τ_j. The lookup is simplified by only considering the average transition slope $\overline{s_j}$ of all switching inputs during an event combination which is reasonable for well designed circuits. The logic states l_i of all N inputs at the beginning of the event combination form the combined decimal equivalent state $L_j = \sum_{i=0}^{N-1} l_i \cdot 2^i$. Each event combination used during library lookup can therefore be written as

$$K_j = \left[\tau_j, \overline{s_j}, L_j, \left(\left(w_0, t'_0\right), \ldots, \left(w_{N-1}, t'_{N-1}\right)\right)\right] \quad (1)$$

where the last parameter is a list of tuples specifying the pulse width w_i and event time offset t'_i for each of the N inputs. For the example depicted in Fig. 3, the event combination with $N = 2$ inputs features a combination time offset $\tau_0 = \min\{t_{0,0}, t_{1,0}\} = 0.1$ with $t'_{0,0} = t_{0,0} - \tau_0 = 0$ and $t'_{1,0} = t_{1,0} - \tau_0 = 0.05$. The average transition slope for the example is calculated as $\overline{s_0} = \frac{1}{2}(s_{0,0} + s_{1,0}) = 28.5$ and the combined input state is $L_0 = l_{0,0} \cdot 2^0 + l_{1,0} \cdot 2^1 = 1$ which results in $K_0 = [0.1, 28.5, 1, ((-1, 0), (0.06, 0.05))]$, where the units of all times and slopes are omitted for the sake of brevity.

C. Library Lookup and Concatenation of Waveforms

The effect of each event combination K_j on the supply current as well as on the gate output voltage can be looked up from the characterization library. Both effects are stored as piecewise linear approximation waveforms where the position and number of supporting points is optimized for accuracy and storage size. In addition to the event combination parameters (with exception of the time offset τ_j), both waveforms exhibit a strong dependency on the capacitive load C_{load} at the gate output which forms an additional lookup parameter.

The waveform segments at the gate output and the supply current that correspond to the event combinations need to be superimposed in order to obtain the full waveforms spanning the whole estimation time. First, all looked-up waveform segments are moved in time according to the offset τ_j corresponding to the respective event combination K_j. In case the resulting segments are nonoverlapping in time, the output and supply current waveforms can be obtained by simple concatenation of the segments. For closely spaced events it is possible for consecutive waveform segments to overlap in selected time windows. In that case the output waveform is constructed using the first intersection of both waveforms as the time where control over the output state switches from one segment to the next.

Due to this concatenation of analog signal waveforms the estimation retains full information about glitch creation and propagation. In order to perform analysis for gates in subsequent estimations steps, the output voltage waveform is analyzed in the same way as the gate inputs before, resulting in a number of event tuples (l, w, t, s), consisting of the logic state l at the beginning of the event, the potential pulse width w, the event time t and the signal slope s. These events are stored in the local signal database until needed in subsequent estimations. The supply current waveform on the other hand can be integrated in order to obtain the power consumption of the current gate.

V. Characterization Complexity

The generation of a sufficiently accurate characterization library is a one-time cost but nevertheless requires some attention in order not to restrict the overall accuracy and keep the storage size of the library in manageable limits. In the following, the operator $\mathrm{n}(\cdot)$ denotes the number of possible parameter

values. The number of distinct combinations to characterize for a gate with N inputs that drives a capacitive output load C_{load} can be given as

$$n_{\text{comb}} = 2^N \cdot \text{n}(C_{\text{load}}) \cdot \text{n}(\overline{s}) \cdot \big[(\text{n}(w) \cdot \text{n}(t) + 1)^N - (\text{n}(w) \times (\text{n}(t) - 1) + 1)^N + 1\big]. \quad (2)$$

This exponential growth of combinations severely limits the number of pulse widths and event times that could be characterized in reasonable time for gates with large number of inputs. The proposed estimation approach limits the number of inputs n_{sw} that are assumed to feature quasisimultaneous switching events. This simplification can be motivated by the reasoning that it becomes increasingly improbable for larger number of inputs to switch at the same time. This way, the number of event combinations to characterize for a gate with $N \geq n_{\text{sw}}$ inputs is reduced to

$$n_{\text{comb,n}_{\text{sw}}} = 2^N \,\text{n}(C_{\text{load}})\,\text{n}(\overline{s}) \times \sum_{k=1}^{n_{\text{sw}}} \binom{N}{k} \Big(\text{n}(w)^k\,\text{n}(t)^k - \text{n}(w)^k(\text{n}(t) - 1)^k\Big) \quad (3)$$

which limits the inner term to cubic complexity for an empirically selected $n_{\text{sw}} = 3$. Each combination is simulated analogous to conventional library characterization using an automated process. The resulting waveforms can be compressed using the same base curve technology that is employed for compact-CCS models which already require analysis and storage of current waveforms [10]. Although the number of switching events to characterize is increased by up to four orders of magnitude, a large number of combinations result in either highly similar output waveforms or negligible output switching. This can be exploited for intelligent selection of switching combinations to characterize and for further reductions of the storage requirements. Without employing these compression techniques, the prototype implementation results in a characterization library that is approximately 800 times larger than a conventional library. Base curve technology alone could reduce this size by a factor of 3–5 without affecting accuracy [10]. As with conventional library characterization, the process can be parallelized to arbitrary degrees.

VI. Demonstration of Estimation Flow

As a demonstration of the proposed power estimation based on waveform lookup, the estimation workflow for a single gate will be examined in detail. The signals shown in Fig. 4(a) are assumed to be connected to a 4-input XNOR gate. These waveforms could form primary inputs to the circuit or could result from a previous estimation iteration. Prior to the actual estimation all waveforms are decomposed into switching events, that can be written as 4-tuples as defined before. In the next step, combinations of events within a maximum distance of δ_t are formed. For the example in Fig. 4(a), the switching events of the individual inputs can be separated into three intervals. This results in four event combinations because the second switching interval needs to be separated into two event combinations to satisfy the constraint chosen during library characterization that a maximum of three inputs would switch during an event combination. Prior to library lookup all event combinations are shifted in time to derive the event offset τ_j. The combined input state L_j at the start of the event combination is calculated and

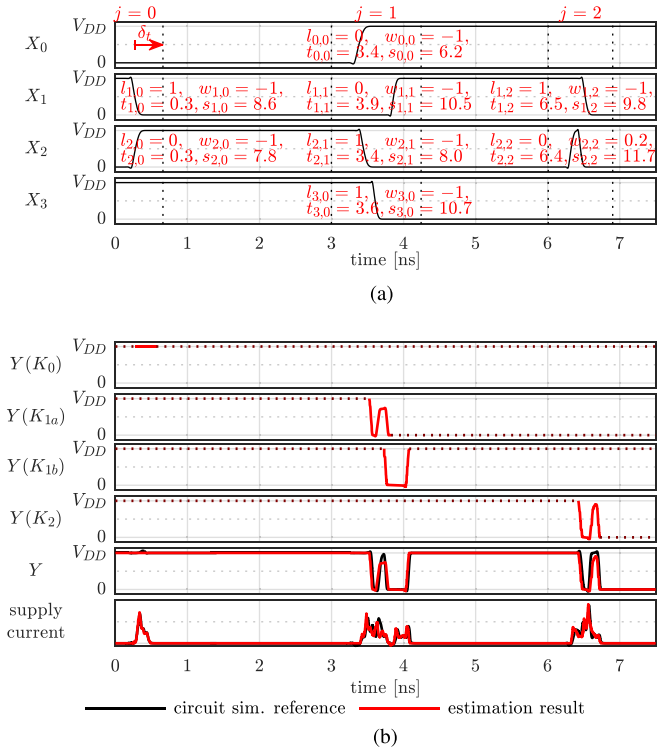

Fig. 4. Waveform lookup demonstration for a 4-input XNOR gate. (a) Input signals to the XNOR gate with event characteristics. (b) Output effects of all event combinations and resulting superposition.

the mean transition slope of all switching inputs $\overline{s_j}$ is derived, which results in

$$K_0 = [0.3, 8.2, 10, ((0, 0), (-1, 0), (-1, 0.01), (0, 0))]$$
$$K_{1a} = [3.4, 8.3, 12, ((-1, 0), (0, 0), (-1, 0.06), (-1, 0.23))]$$
$$K_{1b} = [3.6, 11, 9, ((0, 0), (-1, 0.25), (0, 0), (-1, 0))]$$
$$K_2 = [6.4, 11, 3, ((0, 0), (-1, 0.13), (0.25, 0), (0, 0))].$$

These normalized event combinations in combination with the capacitive output load C_{load}, derived from the netlist and parasitic extraction of interconnects, correspond to the scenarios characterized in the library. This allows lookup of the voltage waveform at the gate output as well as of the supply current waveform for each combination. The first four rows in Fig. 4(b) correspond to the effect on the output waveform of each of the four event combinations which have already been shifted in time by τ_j to reverse the normalization. The estimation result of the output waveform is obtained by concatenation of these looked-up segments. The overlapping effects of K_{1a} and K_{1b} are combined by truncating both segments at the first intersection of both waveforms. In the fifth row of Fig. 4(b), the estimated waveform is compared to the result of a physical-level circuit simulation stimulated by the same input waveforms. After analysis of the switching events on the output node, which results in signal statistics to be used in subsequent logic levels, this voltage waveform is no longer required in the estimation flow. The supply current waveform that can be integrated to obtain the actual power estimation is plotted in the sixth row of the figure. It is constructed from segments related to all event combinations in the same way as the output voltage response and shows the same high level of estimation accuracy.

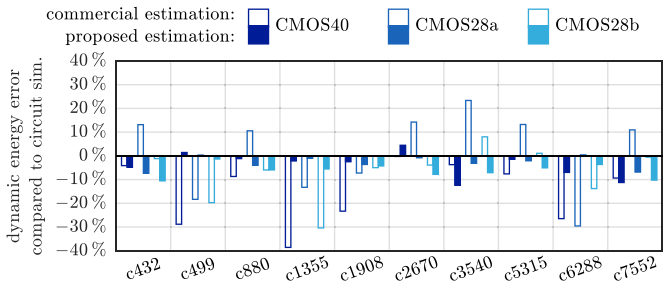

Fig. 5. Comparison of estimation accuracy of proposed event-based waveform lookup compared to state-of-the-art commercial power estimator.

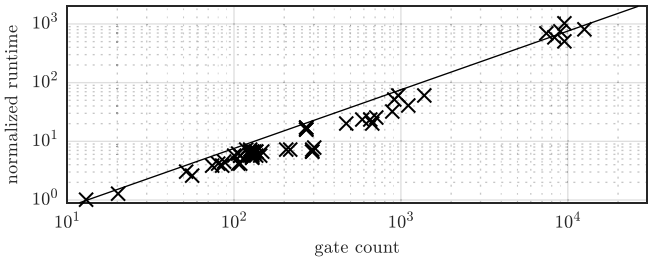

Fig. 6. Runtime of proposed estimation for various circuit sizes.

VII. Estimation Accuracy and Computational Complexity

To evaluate the accuracy of the proposed approach, a range of circuits was analyzed using both the proposed waveform lookup and a commercial gate-level power estimator. The reference power consumption data against which the estimations are compared was generated using circuit simulation. This analysis was carried out using three different advanced commercial CMOS technologies: CMOS40 is a 40-nm CMOS technology while CMOS28a and CMOS28b are both 28-nm CMOS technologies from two different foundries. All benchmark circuits were implemented using the foundry-supplied general purpose standard cell libraries. Fig. 5 compares the accuracy of the proposed approach to that of a state-of-the-art gate-level power estimator for the ISCAS85 benchmark set [12]. The improvements in accuracy result in a mean error of only 4.7% for all circuits and technologies compared to 13.1% for state-of-the-art tools. The maximum error observed for this benchmark set is 12.6% for the proposed approach in contrast to 38.6% for commercial power estimators. The highest gains in accuracy can be observed for circuits containing many XNOR gates (c499/c1355), where multi-input transitions are of critical importance, or deep logic paths of full adders (c6288), which suffer from excessive glitch propagation.

Since the proposed approach traverses the circuit hierarchy iteratively by logic levels and performs power estimation for each logic gate separately, the computational complexity of the algorithm grows linearly with circuit size, similar to conventional gate-level power estimators. Concepts like caching of interpolated waveform results can be applied to further reduce the runtime while the fact that all estimations at the same logic level are uncorrelated lends itself to parallel implementations. The linear runtime complexity of the prototype implementation of the proposed algorithm based on MATLAB is demonstrated in Fig. 6. It outperforms highly optimized commercial circuit simulators by a factor of 50 for circuits with 1000 gates. The speedup increases with growing circuit size as expected. State-of-the-art gate-level power estimators on the other hand exhibit runtimes that are faster by a factor of approximately 200. However, due to similar computational complexity it can be assumed that runtime-optimized implementations of the proposed approach will significantly reduce this gap without sacrificing the superior accuracy.

VIII. Conclusion

A novel power estimation approach is proposed that reduces the inaccuracies of state-of-the-art estimation flows by working on looked-up signal waveform segments. While current gate-level power estimators rely on switching activity information generated in an error-prone preparatory logic simulation or probabilistic propagation, the proposed approach inherently handles propagation of complex switching events. This is achieved by recording the voltage response at the gate output caused by switching on gate inputs during precharacterization in addition to related supply current. The characterization of multi-input events as well as partial-swing glitch pulses allows for accurate estimation of glitch generation and propagation, which has been shown to be a limiting factor for the accuracy of current gate-level power estimators. The estimation results obtained by the proposed approach were demonstrated to reach a high level of accuracy even for circuits suffering from complex signal delay effects that result in significant errors in state-of-the-art estimation methodologies.

References

[1] M. Meixner and T. G. Noll, "Limits of gate-level power estimation considering real delay effects and glitches," in *Proc. Int. Symp. SoC*, Tampere, Finland, 2014, pp. 1–7.

[2] C. M. Huizer, "Power dissipation analysis of CMOS VLSI circuits by means of switch-level simulation," in *Proc. Eur. Solid State Circuits Conf.*, Grenoble, France, 1990, pp. 61–64.

[3] P. Israsena and S. Summerfield, "Novel pattern-based power estimation tool with accurate glitch modeling," in *Proc. Int. Symp. Circuits Syst.*, Geneva, Switzerland, 2000, pp. 721–724.

[4] F. Hu and V. D. Agrawal, "Enhanced dual-transition probabilistic power estimation with selective supergate analysis," in *Proc. Int. Conf. Comput. Design*, San Jose, CA, USA, 2005, pp. 366–369.

[5] Z. Hao *et al.*, "Statistical full-chip dynamic power estimation considering spatial correlations," in *Proc. Int. Symp. Qual. Electron. Design*, Santa Clara, CA, USA, 2011, pp. 1–6.

[6] A. Raghunathan, S. Dey, and N. K. Jha, "High-level macro-modeling and estimation techniques for switching activity and power consumption," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 11, no. 4, pp. 538–557, Aug. 2003.

[7] M. Meixner and T. G. Noll, "Statistical modeling of glitching effects in estimation of dynamic power consumption," in *Proc. Int. Conf. VLSI Design*, Mumbai, India, 2014, pp. 415–420.

[8] X. Liu and M. C. Papaefthymiou, "Incorporation of input glitches into power macromodeling," in *Proc. Int. Symp. Circuits Syst.*, Scottsdale, AZ, USA, 2002, pp. IV-846–IV-849.

[9] *VCS MX User Guide*, Synopsys, Mountain View, CA, USA, 2013.

[10] *Liberty User Guide, Volume 1*, Synopsys, Mountain View, CA, USA, 2013.

[11] M. Dietrich and J. Haase, *Process Variations and Probabilistic Integrated Circuit Design*. New York, NY, USA: Springer, 2012.

[12] F. Brglez and H. Fujtwara, "A neutral netlist of 10 combinational benchmark circuits," in *Proc. IEEE Int. Symp. Circuits Syst.*, Kyoto, Japan, 1985, pp. 677–692.