Efficient System-Level Aging Prediction

Hatami, Nadereh; Baranowski, Rafal; Prinetto, Paolo; Wunderlich, Hans-Joachim

Proceedings of the 17th IEEE European Test Symposium (ETS'12) Annecy, France, 28 May-1 June 2012

doi: http://dx.doi.org/10.1109/ETS.2012.6233028

Abstract: Non-functional properties (NFPs) of integrated circuits include reliability, vulnerability, power consumption or heat dissipation. Accurate NFP prediction over long periods of system operation poses a great challenge due to prohibitive simulation costs. For instance, in case of aging estimation, the existing low-level models are accurate but not efficient enough for simulation of complex designs. On the other hand, existing techniques for fast high-level simulation do not provide enough details for NFP analysis. The goal of this paper is to bridge this gap by combining the accuracy of low-level models with high-level simulation speed. We introduce an efficient mixed-level NFP prediction methodology that considers both the structure and application of a system. The system is modeled at transaction-level to enable high simulation speed. To maintain accuracy, NFP assessment for cores under analysis is conducted at gate-level by cycle-accurate simulation. We propose effective techniques for cross-level synchronization and idle simulation speed-up. As an example, we apply the technique to analyze aging caused by Negative Bias Temperature Instability in order to identify reliability hot spots. As case studies, several applications on an SoC platform are analyzed. Compared to conventional approaches, the proposed method is from 7 up to 400 times faster with mean error below 0.006%.

Efficient System-Level Aging Prediction

Nadereh Hatami*†, Rafal Baranowski*, Paolo Prinetto† and Hans-Joachim Wunderlich*
*University of Stuttgart, Institute of Computer Architecture and Computer Engineering
Pfaffenwaldring 47, D-70569 Stuttgart, Germany
†Politecnico di Torino, Dipartimento di Automatica e Informatica
Corso Duca degli Abruzzi 24, I-10129 Torino TO, Italy

***Abstract*—Non-functional properties (NFPs) of integrated circuits include reliability, vulnerability, power consumption or heat dissipation. Accurate NFP prediction over long periods of system operation poses a great challenge due to prohibitive simulation costs. For instance, in case of aging estimation, the existing low-level models are accurate but not efficient enough for simulation of complex designs. On the other hand, existing techniques for fast high-level simulation do not provide enough details for NFP analysis.**

The goal of this paper is to bridge this gap by combining the accuracy of low-level models with high-level simulation speed. We introduce an efficient mixed-level NFP prediction methodology that considers both the structure and application of a system. The system is modeled at transaction-level to enable high simulation speed. To maintain accuracy, NFP assessment for cores under analysis is conducted at gate-level by cycle-accurate simulation. We propose effective techniques for cross-level synchronization and idle simulation speed-up. As an example, we apply the technique to analyze aging caused by Negative Bias Temperature Instability in order to identify reliability hot spots. As case studies, several applications on an SoC platform are analyzed. Compared to conventional approaches, the proposed method is from 7 up to 400 times faster with mean error below 0.006%.

***Index Terms*—Non-functional properties, Transaction Level Modeling (TLM), mixed-level simulation, aging analysis, Negative Bias Temperature Instability (NBTI)**

I. Introduction

With the increasing complexity of embedded systems, non-functional aspects become as important as functionality. Power consumption and heat dissipation – the most prominent non-functional parameters – enforced power-aware methodologies for design and verification. As the scaling of technology nodes proceeds further, aging processes, such as the effect of negative bias temperature instability (NBTI), arise as another non-functional bottleneck that requires consideration in early design phases [1–3].

High accuracy is achieved by modeling non-functional parameters at low abstraction levels. Existing models for power consumption and heat distribution rely on the proportional relation of transistor switching activity and heat dissipation [4, 5]. Models for robustness and vulnerability take into account electrical and logical fault masking at gate- and transistor-level [6, 7]. Models for aging mechanisms such as NBTI and Hot Carrier Injection (HCI) are governed by transistor-level workload [8, 9].

All the above-mentioned NFP models require that certain gate- or transistor-level observables be available throughout system operation. For instance, for power and HCI aging models, the transistor switching activity is an essential observable. For vulnerability and NBTI aging models, the gate-level input patterns are required.

Over long simulation periods, a system-wide analysis with fine grain NFP models is computationally expensive, if feasible at all. As the accurate NFP models require acquisition of fine grain observables, they are not scalable to large systems and complex applications.

As non-functional parameters influence each other and are interrelated, accurate estimation of non-functional parameters requires a holistic simulation methodology that considers the structure of a system, its functionality, and the target application. Throughout simulation, all relevant system observables must be captured and the NFPs evaluated on-the-fly.

In existing techniques for NFP prediction, the workload of a design under analysis (DUA) is usually modeled by input signal probabilities or by a set of "typical workload patterns" [9–11] that reflect the average application. As the technology scales, the amplitude of NFP fluctuations under various workloads (e.g. worst vs. average case) is growing [12], which necessitates either a pessimistic worst-case analysis, or an extensive simulation of the target application.

The goal of this work is to enable an accurate NFP evaluation of a DUA in the target system and for the target application. The system and its application are modeled at transaction-level to achieve high simulation efficiency. Transistor-level observables, as required by the fine grain NFP models, are captured in gate-level simulation.

The concept of mixed-level modeling has found widespread use for design validation [13], fault simulation [14], as well as prediction of non-functional properties such as power [15]. This work does not aim at competing with such specialized approaches, but rather at providing the basis for NFP prediction using mixed-level simulation up to the transaction level. This paper presents, for the first time, a universal framework that combines transaction- and gate-level models for efficient acquisition of transistor-level NFP observables. The proposed approach benefits from high simulation speed of abstract models, while gate-level simulation is accelerated by the presented technique of simulation fast forwarding. As a result, the proposed method yields precise NFP observables at drastically reduced simulation cost.

As gate-level models are required only for the components under NFP analysis, the proposed method can be used early in the design flow for evaluating the design alternatives w.r.t. non-functional properties. To this end, high-level system models from design space exploration can be reused.

As a proof of concept, we show the application of the proposed method to acquisition of observables required for NBTI aging prediction. Although we apply the proposed method to NBTI prediction, it can well be used for other non-functional parameters such as power, vulnerability, or HCI aging.

The paper is organized as follows: The proposed method is discussed in section 2. Section 3 presents its application to NBTI aging estimation, followed by experimental results in section 4.

II. Methodology

This section presents the proposed approach which combines high-level system simulation with low-level NFP evaluation. An efficient procedure for mixed-level simulation and monitoring of low-level observables is described.

An overview of the proposed method is shown in figure 1. The high-level workload of the design under analysis (DUA) is used for low-level NFP analysis. During gate-level simulation, relevant NFP observables are captured.

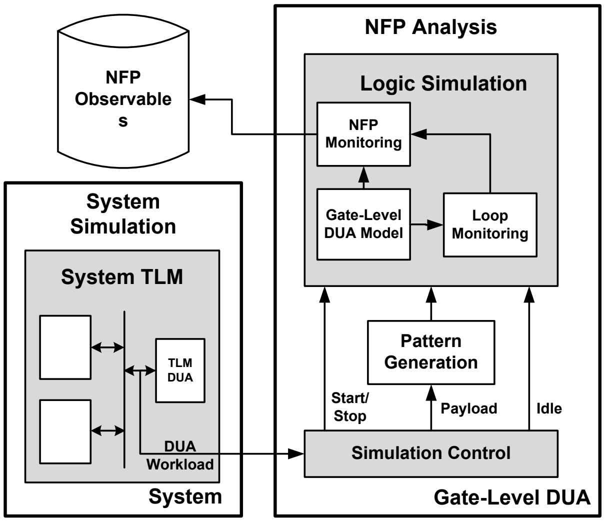

Fig. 1. Overview of the mixed-level simulation approach

A. High-level system simulation

To enable high-speed simulation, the temporal and functional behavior of system hardware and software modules is modeled at transaction-level. Transaction-level system models (TLM's) are often used in design space exploration, and can be reused here for the purpose of NFP evaluation.

TLM allows flexibility in temporal modeling [16], while the accuracy can be dynamically adapted during simulation [17]. In loosely-timed transaction-level models (TLM), the temporal behavior is managed at the level of transactions. In the more accurate approximately-timed TLM, the time is managed at sub-transaction level, i.e., it is updated for each transaction phase, such as request, transfer, and acknowledgment. While the loosely-timed TLM is faster to simulate, the approximately-timed TLM provides finer grain estimation of system timing, which is desirable for accurate NFP analysis.

During high-level system simulation, the workload of the transaction-level DUA model is continuously monitored. Whenever the DUA receives a transaction request, the payload of the transaction and its time-stamp are captured as *high-level DUA workload*. The high-level DUA workload is subject to NFP analysis as described below.

B. Low-level DUA simulation

The DUA is modeled at low-level to enable accurate analysis of NFP observables. The low-level DUA model is a post-synthesis gate-level netlist, consisting of primitive gates and registers.

The low-level DUA model is simulated using the high-level DUA workload as stimuli. To this end, the transaction payload is automatically translated into pin- and cycle-accurate input patterns. The gate-level simulation of the high-level workload is conducted until the transaction is acknowledged by the DUA, e.g. by setting an acknowledge signal.

For the sake of brevity, we assume that the gate-level model is cycle-accurate. If models with post-synthesis timing annotation are available, the proposed approach can be extended with accurate gate-level timing simulation.

C. Mixed-level simulation procedure

Concurrent simulation of models at different abstraction levels requires proper simulation control. In the following, we explain the mixed-level simulation procedure.

Initially, just the high-level system simulation is run. As long as the transaction-level DUA model is not involved in simulation, the gate-level DUA simulation is deferred. Upon detection of a transaction, the high-level DUA workload at time t_0 is sent for low-level simulation. Before the transaction is processed at gate-level, the idle time that passed since the last transaction up to t_0 is simulated first. The idle simulation is required to synchronize the state of the gate-level model with the simulation time at transaction-level. After the simulation time at gate-level reaches t_0, the simulation of the transaction begins. Input patterns are automatically generated, and the gate-level simulation proceeds as long as is required to complete the transaction. After the DUA provides an acknowledge signal, the gate-level simulation is stopped until another high-level DUA workload is available.

In order to synchronize the simulation of high- and low-level models and prevent data loss, a queue is used to store the incoming high-level workload. The high-level simulation is not blocked by the low-level analysis. An example is given in figure 2: The high-level simulation of the second request is processed without delay, as the transaction-level DUA model provides a fast response to the system, and the workload is queued for gate-level simulation.

D. Accelerated idle simulation

Non-functional parameters need to be evaluated both in the active operation mode, when the core is busy processing a transaction request, as well as when it is idle. In the following, we present a technique to accelerate the idle simulation by fast forwarding.

Before processing an incoming transaction, the gate-level simulation is run for the idle time that passed since the last

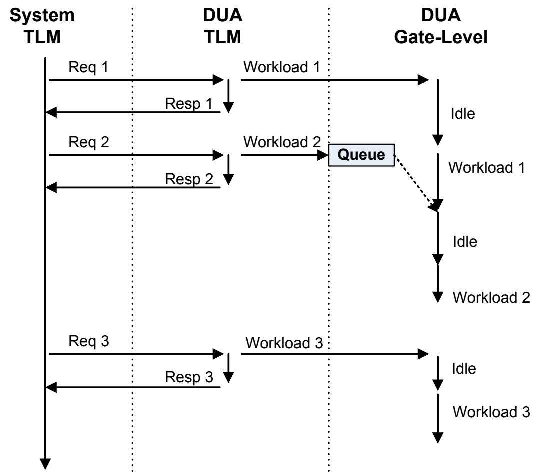

Fig. 2. Cross-level synchronization example

transaction. During this time, the DUA may perform internal operations but it does not receive any incoming transactions. The idle simulation is therefore performed with primary inputs kept constant. In idle simulation, the sequential state of the DUA (content of its constituent registers) is either stable or it follows a loop pattern, as shown in figure 3. In the following, an efficient simulation procedure is explained that allows to skip (fast forward) the idle simulation if a sequential idle loop is found.

The gate-level simulation procedure is shown in figure 4. Upon reception of a transaction at time t_0, the simulation is advanced as follows:

Let t_G be the actual simulation time of the gate-level model, and let S be a sequence of states, which is initialized with the current DUA state at t_G. The DUA state is determined by the content of all registers within the DUA gate-level model.

The idle period of length $t_0 - t_G$ is simulated first, with DUA primary inputs kept constant. In each simulation cycle, the current DUA state is checked against S. If a match is found, an idle loop is detected. If there is no match, the current DUA state is appended to S, and the next idle cycle is simulated.

If an idle loop of length T_L has been found, the simulation does not proceed cycle-accurately. Instead, the simulation time is fast-forwarded up to time $t_G + \lfloor(t_0 - t_G)/T_L)\rfloor \cdot T_L$, effectively skipping the simulation of $\lfloor(t_0 - t_G)/T_L)\rfloor$ identical idle loop iterations.

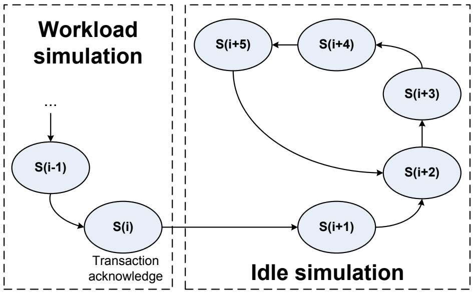

Fig. 3. Example of an idle sequential loop

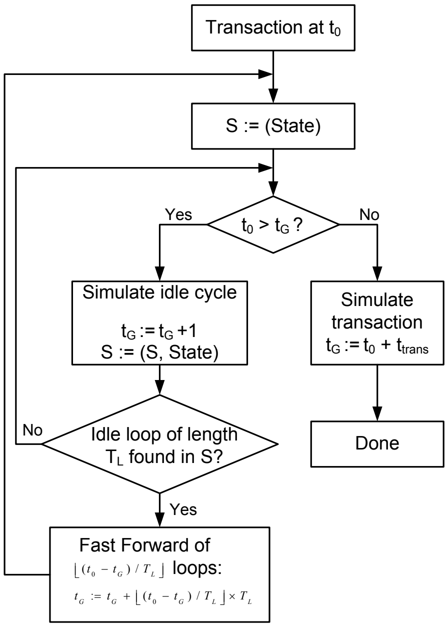

Fig. 4. Gate-level simulation procedure

Finally, after the gate-level simulation time has reached the current transaction time $(t_0 = t_G)$, the transaction workload is simulated. The gate-level simulation is stopped as soon as the transaction is acknowledged, and it remains on hold until the next request comes.

The proposed procedure requires that in each idle simulation cycle the DUA state is captured and compared to all the states that have previously been stored. The idle loop detection can be conducted in parallel to the gate-level simulation. Both the computational effort and the required storage space is confined by setting a limit for the idle loop depth. If the actual idle loop exceeds this limit, the loop is not detected and the idle simulation proceeds cycle-accurately.

E. Collecting NFP observables

During gate-level simulation, the internal design nodes (gate and register inputs) are continuously monitored. All simulation events (observables) that are relevant for NFP analysis are evaluated and stored. The collected observables constitute an interface of the proposed method, providing accurate input to analytical NFP models.

For power and HCI aging estimation, signal switching activity is the required observable. For NBTI aging prediction, the operation mode of PMOS transistors is captured, as discussed in the following section.

For linear analytical NFP models, the observables are collected simply by counting the number of observed events. E.g., for linear power models, the cumulative number of signal transitions is collected for each node.

Special care must be taken in case of non-linear NFP models that require the temporal behavior of an observable as an input. For instance, the reaction-diffusion NBTI model [18] requires continuous observation of the transistor operation mode. To reduce the effort of observable acquisition and storage, sampling or approximation techniques can be applied.

Due to simulation fast forwarding, the acquisition of observables in idle simulation requires special attention. The observables are evaluated by simulating the idle loop only once. For linear NFP models, the acquired observables are simply taken with an appropriate weight. Non-linear NFP models may require interpolation, which is beyond the scope of this work.

III. Application to aging prediction

As a proof of concept, we apply the proposed method for NBTI aging estimation. The NBTI effect results in a gradual positive shift of the threshold voltage of PMOS transistors when they operate in inversion [2, 8, 9], causing a gradual degradation of the overall circuit performance [19].

As the degradation is a dynamic process characterized by stress and recovery phases, the workload of transistors is a crucial observable [3, 18, 20, 21]. For the sake of brevity, in the following we consider simplified NBTI models, in which the degradation is a function of the cumulative time over which the transistor has been in inversion [10, 19, 22].

For a transistor c, we define the stress factor $a_c(t)$ as the cumulative time the transistor has been exposed to inversion until time t. In the following, we show the application of the proposed method for the acquisition of this observable.

A. Evaluation of stress factors

The stress factors for individual transistors are obtained through cycle-accurate simulation of the gate-level model. Each gate g is associated with a stress table S_g with 2^i integer entries, where i is the number of inputs to g. The entries in S_g correspond to the possible input patterns, e.g. $S_g(0)$ stores the number of times the pattern 00 has been observed at gate g with two inputs.

In each simulation cycle, for every gate g exactly one entry in stress table S_g is incremented, according to its actual input pattern.

Due to simulation fast forwarding, the evaluation of the stress exerted in idle simulation requires special handling. If an idle sequential loop is found, it is simulated again to determine the incremental stress that it causes to stress tables. The incremental stress is then multiplied by the number of skipped loop iterations, and added to the stress tables.

After the simulation is completed, the stress factors for individual transistors are simply derived from the stress tables. For instance, for a CMOS inverter i, the stress factor of its PMOS transistor is equal to the content of $S_i(1)$ divided by the clock frequency. For a PMOS transistor in a more complex gate, the stress factor is calculated as a sum of those entries in the respective stress table that correspond to the patterns putting the transistor in inversion.

B. Discussion of accuracy

The proposed method has two underlying assumptions:

1) The inputs to the DUA are stable, except when the DUA receives a transaction.
2) The high-level system model accurately predicts the temporal system behavior.

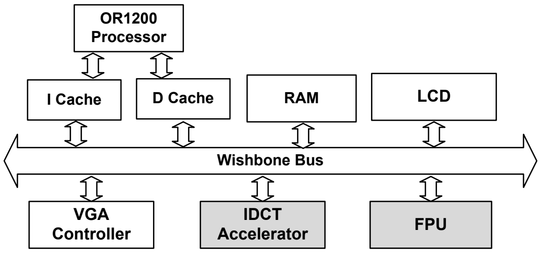

Fig. 5. System architecture

The first assumption holds for many point-to-point communication schemes, but is not valid in general. In a bus-centric system, the DUA bus inputs are exposed to switching activity generated by other system components. This activity may affect the gates in the combinational transitive fan-out of bus inputs. The incurred inaccuracy is marginal if all the interface inputs are registered or gated, which is a standard design practice [23].

The second assumption is justified as long as the temporal behavior of the high-level DUA model matches its actual hardware operation, i.e., the time-stamps of the high-level DUA workload are accurate. In general, an exact match cannot be achieved due to the abstract nature of the system model. The impact of this modeling inaccuracy is discussed in the experimental section.

C. Implementation

The NFP simulation framework consists of two parts:

- Aging analyzer (Java)
- System simulator (SystemC)

The system is implemented at transaction-level and simulated by the SystemC simulation kernel. During simulation, the high-level DUA workload in form of time-annotated transaction payloads is sent to the aging analyzer via TCP/IP protocol.

The aging analyzer is implemented in Java. It contains a gate-level model of the DUA, performs pattern generation and gate-level logic simulation. The stress factors are monitored cycle-accurately.

The framework supports multiple DUAs. The aging analyzers work in parallel and can be distributed across different machines. The system simulator communicates the workload of a DUA to the respective aging analyzer.

IV. Case Study

The proposed method is applied to the case study of an SoC platform based on a 32-bit Wishbone bus and an OpenRISC-1200 processor[1]. As depicted in figure 5, the system is equipped with two display controllers (VGA and LCD), an Inverse Discrete Cosine Transform accelerator (IDCT), and a floating point unit (FPU). The processor communicates with the peripheral devices through the Wishbone bus using single 32-bit read/write bus cycles.

The subject of the case study is the aging analysis of the IDCT accelerator and the FPU. The accuracy and performance of the proposed method is evaluated in several validation experiments.

[1] OpenRISC Project, http://opencores.org/or1k

A. Experimental Setup

The system is modeled functionally at transaction-level using the SystemC language and the TLM-2.0 modeling library. Each functional unit is modeled as a set of concurrent processes representing its behavior. The units are approximately-timed and communicate via TLM sockets.

The IDCT accelerator has been obtained from OpenCores[2]. It is a sequential implementation of the IDCT algorithm with integer precision. The core has been extended with two FIFO memories for input and output data buffers, and a slave interface to the Wishbone bus.

The FPU has been extracted from the MicroSparcII processor[3]. It is a multi-cycle floating point unit of double precision. The core has been equipped with a Wishbone slave interface.

To obtain structural gate-level models, the two DUAs were synthesized for the LSI10k generic library. The IDCT accelerator consists of 28,772 gates (113,686 transistors in CMOS technology) and 3,775 registers. The FPU requires 17,113 gates (62,362 transistors) and 630 registers.

B. Applications

The IDCT accelerator has been evaluated in several applications with JPEG image decompression [24]. The processor periodically decodes and displays a color image. The process of Huffman decoding, dequantization, and the final color conversion is performed purely in software, while the IDCT transformation is offloaded to the IDCT accelerator. The IDCT core performs the transformation on 8x8 pixel blocks.

The second type of applications takes advantage of the FPU. The processor runs several signal processing applications:

- FIR: high-pass Finite Impulse Response filter of order 14
- IIR: Low-pass Butterworth filter of order 5
- FFT: Fast Fourier Transform for vectors of length 64
- IFFT: Inverse FFT (vector length 64)
- JPEG: image decompression with a floating point IDC transformation for an 8x8 pixel image.

Floating point operations, such as addition, multiplication, and division, are offloaded to the FPU. As input, random data (noise) is used.

C. Validation Experiments

For validation, the system is modeled at register transfer level (RTL). The system model is simulated using a commercial tool. The bus traffic at the interface to the IDCT accelerator and the FPU is captured in form of a value change dump (VCD file). The structural gate-level model of the DUA (either the IDCT accelerator or the FPU) is then simulated in the Java aging simulator using the VCD file as stimulus.

The applications that are run in the validation experiment are equivalent to those analyzed with the proposed method (the processor runs the very same software). In the proposed method the system is modeled at transaction-level and fast-forwarding is used to improve gate-level simulation performance. For the validation, a cycle-accurate RTL system model is used, and the gate-level simulation is run cycle-accurately without fast forwarding.

For transistor c, the reference stress factor $a_c(t)$ is acquired from the validation experiment, while its estimation $s_c(t)$ is provided by the proposed method. The absolute estimation error is:

$$e_c(t) = a_c(t) - s_c(t)$$

The accuracy of the proposed method is measured as a relative mean error over all constituent transistors $c \in \mathbb{C}$ at the end of simulation time $t = T$:

$$Error = \frac{\sum_{c \in \mathbb{C}} |e_c(T)|}{T \cdot |\mathbb{C}|}$$

D. Results for the IDCT Core

The IDCT core is evaluated in several JPEG decompression applications with images of size from 8x8 up to 256x256 pixel, as shown in table I. The application lengths in clock cycles are given in the second column. The third column "Prediction time" gives the time required by the proposed approach to calculate the stress factors. Column 5, "Validation time" gives the time required by the validation experiment for system simulation at RTL, and DUA aging simulation at gate-level (GL). The speed-up of the proposed method is evaluated with respect to the total validation time at RT- and gate-level. The proposed prediction method is faster than the validation experiment from 205 up to 436 times. The mean error is stable at 0.002%. This error results mostly from the inaccuracy of temporal modeling at system-level: High abstraction level results in a slight mismatch of the total application length (and the timing of DUA workload) in transaction- and RT-level simulation.

Image	App.	Pred.	Validation		Speedup	Error
size	length	time	time			
	[cyc.]	[s]	RTL [h]	GL [h]	[x]	[%]
8x8	2.67 M	82	0.07	4.6	205	0.002
16x16	3.74 M	83	0.13	9.4	411	0.002
64x64	32.80 M	904	0.75	109.0	436	0.002
256x256	354.00 M	11245	12.20	1323.0	427	0.002

TABLE I
JPEG DECOMPRESSION: PERFORMANCE AND ACCURACY

E. Results for the FPU Core

The FPU core is evaluated in several signal processing applications, as shown in table II. Compared with the previously discussed JPEG applications, these applications use the DUA more intensively, as the majority of CPU operations is offloaded to the FPU. For this reason, the performance gain due to simulation fast forwarding is less compared to the previous results. Nevertheless, the proposed method considerably reduces the effort of gate-level simulation, and achieves a minimal speed-up of 7.6x with an average error below 0.006%.

[2] http://www.opencores.org

[3] Oracle,http://www.oracle.com/us/sun

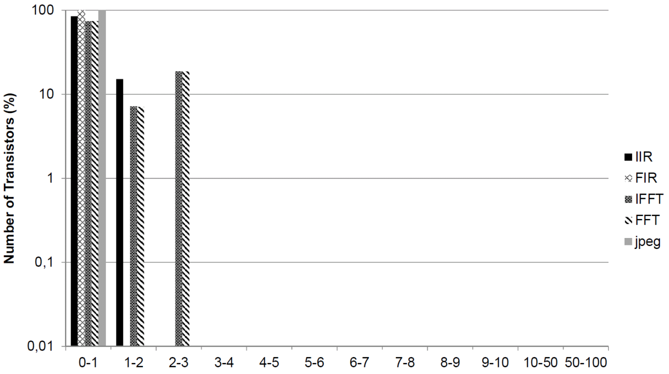

Fig. 6. FPU applications: error histogram

	App.	Pred.	Validation		Speedup	Error
App.	length	time	time			
	[cyc.]	[h]	RTL [h]	GL [h]	[x]	[%]
IIR	3.7 M	0.37	0.17	3.4	9.7	0.003
FIR	18.5 M	1.00	0.55	7.7	8.2	0.001
FFT	24.4 M	0.72	0.58	7.6	11.3	0.005
IFFT	217.0 M	5.59	4.59	103.0	19.3	0.006
JPEG	121.0 M	10.63	3.46	77.4	7.6	0.002

TABLE II
FPU APPLICATIONS: PERFORMANCE AND ACCURACY

Figure 6 shows a histogram of stress factor estimation errors evaluated for the FPU applications. In all scenarios, more than 70% of stress factors are estimated with an error below 1%. The maximum estimation error is below 3%. This error is observed for the transistors that constitute the gates of the transitive combinational fan-in of the Wishbone bus. This error results from the assumption that the primary DUA inputs are stable during idle simulation, as discussed in section III-B.

V. Conclusions

We proposed an efficient approach for non-functional property analysis using mixed-level simulation. High simulation performance is achieved with transaction-level simulation, while good precision is enabled by gate-level analysis. We presented a technique to reduce the effort of gate-level simulation by simulation fast forwarding. As an example, the application of the method to NBTI aging prediction is shown. The proposed method can also be used for accurate evaluation of other aging effects, power consumption, or vulnerability. Experimental results show that the proposed technique enables a considerable simulation speed-up with marginal influence on simulation accuracy.

Acknowledgment

This work has been supported by the DFG Project OASIS under grant WU 245/11-1.

References

[1] M. Alam and S. Mahapatra, “A comprehensive model of PMOS NBTI degradation,” *Microelectronics Reliability*, vol. 45, no. 1, pp. 71–81, 2005.

[2] T. Kim, R. Persaud, and C. Kim, “Silicon odometer: an on-chip reliability monitor for measuring frequency degradation of digital circuits,” *IEEE Journal of Solid State Circuits*, vol. 43, no. 4, pp. 874–880, 2008.

[3] J. Keane, T.-H. Kim, and C. Kim, “An On-Chip NBTI Sensor for Measuring pMOS Threshold Voltage Degradation,” *IEEE Trans. VLSI Systems*, vol. 18, no. 6, pp. 947 –956, 2010.

[4] A. Ghosh, S. Devadas, K. Keutzer, and J. White, “Estimation of average switching activity in combinational and sequential circuits,” *Proc. Design Automation Conference (DAC)*, pp. 253–259, 1992.

[5] W. Huang, M. R. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusam, “Compact thermal modeling for temperature-aware design,” in *Proc. Design Automation Conference (DAC)*, 2004, pp. 878–883.

[6] S. Mukherjee, C. Weaver, J. Emer, S. Reinhardt, and T. Austin, “A systematic methodology to compute the architectural vulnerability factors for a high-performance microprocessor,” in *Proc. Int’l Symposium on Microarchitecture (MICRO)*, 2003, pp. 29–40.

[7] F. Wang and Y. Xie, “Soft error rate analysis for combinational logic using an accurate electrical masking model,” *IEEE Trans. Dependable and Secure Computing*, vol. 8, no. 1, pp. 137 –146, jan.-feb. 2011.

[8] Y. Wang, H. Luo, K. He, R. Luo, H. Yang, and Y. Xie, “Temperature-aware NBTI modeling and the impact of input vector control on performance degradation,” in *Proc. Design, Automation Test in Europe (DATE) Conference*, 2007, pp. 546–551.

[9] W. Wang, Z. Wei, S. Yang, and Y. Cao, “An efficient method to identify critical gates under circuit aging,” in *Proc. Int’l Conference on Computer-Aided Design (ICCAD)*, 2007, pp. 735–740.

[10] D. Lorenz, G. Georgakos, and U. Schlichtmann, “Aging analysis of circuit timing considering NBTI and HCI,” in *Proc. Int’l On-Line Testing Symposium (IOLTS)*, 2009, pp. 3–8.

[11] W. Wang, S. Yang, S. Bhardwaj, S. Vrudhula, F. Liu, and Y. Cao, “The Impact of NBTI Effect on Combinational Circuit: Modeling, Simulation, and Analysis,” *IEEE Trans. VLSI Systems*, vol. 18, no. 2, pp. 173–183, feb. 2010.

[12] J. Srinivasan, S. Adve, P. Bose, and J. Rivers, “The impact of technology scaling on lifetime reliability,” in *Proc. Int’l Conference on Dependable Systems and Networks (DSN)*, 2004, pp. 177–186.

[13] P. Paulin, C. Pilkington, and E. Bensoudane, “StepNP: a system-level exploration platform for network processors,” *IEEE Trans. Design & Test of Computers*, vol. 19, no. 6, pp. 17–26, nov/dec 2002.

[14] R. Baranowski, S. Di Carlo, N. Hatami, M. Imhof, M. Kochte, P. Prinetto, H. Wunderlich, and C. Zoellin, “Efficient multi-level fault simulation of HW/SW systems for structural faults,” *SCIENCE CHINA Information Sciences*, vol. 54, no. 9, pp. 1784–1796, 2011.

[15] D. Brooks, V. Tiwari, and M. Martonosi, “Wattch: A Framework for Architectural-Level Power Analysis and Optimizations,” *SIGARCH Comput. Archit. News*, vol. 28, pp. 83–94, May 2000.

[16] F. Ghenassia, Ed., *Transaction-Level Modeling with SystemC - TLM Concepts and Applications for Embedded Systems*. Springer, 2005.

[17] M. Radetzki and R. S. Khaligh, “Accuracy-adaptive simulation of transaction level models,” in *Proc. Design, Automation and Test in Europe (DATE)*, 2008, pp. 788–791.

[18] G. Chen, K. Chuah, M. Li, D. Chan, C. Ang, J. Zheng, Y. Jin, and D. Kwong, “Dynamic NBTI of PMOS transistors and its impact on device lifetime,” in *Proc. Int’l Reliability Physics Symposium*, 2003, pp. 196–202.

[19] S. Kumar, C. Kim, and S. Sapatnekar, “An Analytical Model for Negative Bias Temperature Instability,” in *Proc. Int’l Conference on Computer-Aided Design (ICCAD)*, 2006, pp. 493–496.

[20] W. Wang, S. Yang, S. Bhardwaj, R. Vattikonda, S. Vrudhula, F. Liu, and Y. Cao, “The Impact of NBTI on the Performance of Combinational and Sequential Circuits,” in *Proc. Design Automation Conference (DAC)*, 2007, pp. 364–369.

[21] W. Wang, V. Reddy, A. Krishnan, R. Vattikonda, S. Krishnan, and Y. Cao, “Compact Modeling and Simulation of Circuit Reliability for 65-nm CMOS Technology,” *IEEE Trans. Device and Materials Reliability*, vol. 7, no. 4, pp. 509–517, dec. 2007.

[22] S. Bhardwaj, W. Wang, R. Vattikonda, Y. Cao, and S. Vrudhula, “Predictive Modeling of the NBTI Effect for Reliable Design,” in *Proc. Custom Integrated Circuits Conference (CICC)*, 2006, pp. 189–192.

[23] M. Keating and P. Bricaud, *Reuse methodology manual for system-on-a-chip designs*. Springer, 2002.

[24] G. Wallace, “The JPEG Still Picture Compression Standard,” *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. 18 –34, feb 1992.