Cross-Layer Approaches for an Aging-Aware Design Space Exploration for Microprocessors

Fabian Oboril, and Mehdi B. Tahoori
Chair of Dependable Nano Computing (CDNC), Karlsruhe Institute of Technology (KIT)
Email: {fabian.oboril, mehdi.tahoori}@kit.edu

***Abstract*—With the continuous scaling of CMOS technologies, maintaining the microprocessor reliability becomes a major design challenge. In particular, accelerated transistor aging is a serious reliability concern, as it considerably reduces the operational system lifetime. To address this issue, in this work cross-layer solutions for aging modeling, simulation and mitigation are proposed, to be able to co-optimize reliability together with the traditional design constraints such as power, performance, and cost. Therefore, the knowledge from several abstraction layers, ranging circuit- to architecture-level, are exploited for cost-effective aging-aware architecture and system design. The comprehensive simulations and experimental analysis performed in this work show the benefits of this approach over state-of-the-art single-layer solutions.**

I. Introduction

Thanks to the aggressive scaling of transistor dimensions in the past decades, computing systems have revolutionized our life. However, in the shade of the downscaling benefits such as increased microprocessor performance, more integrated features and improved energy/cost efficiency, the *reliability* of nanoscale devices became a major threat for the future success of computing systems (see Fig. 1) [2]–[5]. As a result, with every new technology node, it becomes harder for chip manufacturers to ensure the reliable operation of their chips, and thus malfunctions during the operational mode, that can lead to erroneous program outputs or even system crashes, become more likely.

Among various reliability challenges, *accelerated transistor aging* is of particular importance, as it degrades the transistor switching speed, and thus leads to slower circuits over time [3], [6]. Consequently, in synchronous digital systems, timing failures due to the increased circuit delay can occur and cause incorrect system states. Because of that, the microprocessor lifetime and as a result also the overall system lifetime is considerably reduced, if no countermeasures are taken. This is especially critical for embedded systems that require long mission times, for instance in health care (e.g. implants), space missions (e.g. satellites) or electronic control units (e.g. in airplanes) [7]. Therefore, it is a necessity to consider reliability, and in particular lifetime, as another design constraint, beside the traditional performance, power and cost parameters. However, due to the strong [illegible] among these different criteria, the co-optimization

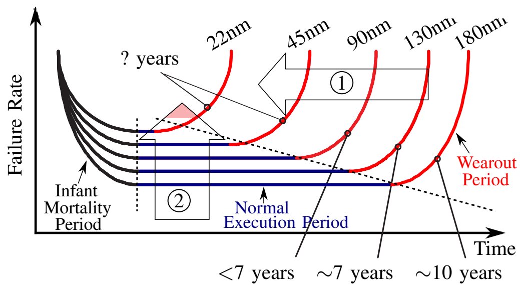


Fig. 1. Increasing unreliability in nanoscale technology nodes due to accelerated transistor aging ① and susceptibility to noise as well as soft errors ② (based on [1])

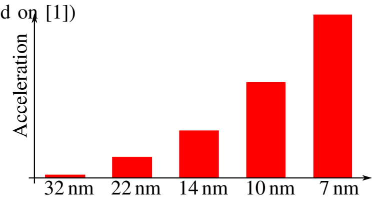


Fig. 2. Aging acceleration in the next technology nodes (based on [2])

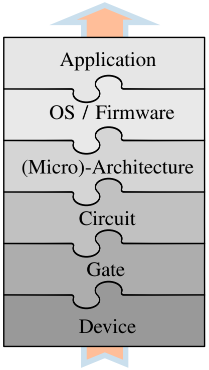


Fig. 3. Abstraction layers in the hardware-software design stack

In addition, lots of effort is spent on improvements at the lowest hardware layers (i.e. at transistor/gate-level), as these layers are very close to the physical origin of the problem (see Fig. 3). However, the influence of higher levels in the hardware-software design stack is neglected in most state-of-the-art solutions, although these layers have a considerable impact on the system lifetime, for example by influencing the thermal behavior of the microprocessor. Therefore, it is crucial to consider the effect of these higher layers, to achieve cost-efficient resilient computer systems. In fact, due to the extend of transistor aging (see Fig. 2) [2], it will be necessary in future that various layers contribute in a combined fashion (i.e. *cross-layer*[1]) to co-optimize lifetime with the other design parameters more efficiently compared to state-of-the-art solutions, which are typically single-layer approaches [9], [10].

In this work we push cross-layer solutions for aging modeling and simulation, as well as aging mitigation forward. Therefore, we address the major transistor aging phenomena that cause a gradual degradation of the device parameters such as switching delay, namely Bias Temperature Instability (BTI) [11] and Hot Carrier Injection (HCI) [12]. In detail, our contributions are as follows:

1) A set of novel *cross-layer aging modeling and analysis frameworks* was developed to allow an effective design space exploration throughout the different microprocessor design phases considering the interdependencies of the different design parameters including lifetime. Compared to existing state-of-the-art solutions the advantage of the proposed frameworks is that a much wider design space can be explored due to the cross-layer approach combined with the architectural aging models which allows to evaluate more parameters. Consequently, using these platforms, the most critical processor components can be identified, and selective, cost-efficient cross-layer aging mitigation techniques can be designed.
2) Using the aforementioned cross-layer platforms a set of *efficient cross-layer aging mitigation techniques* was designed that outperform the existing state-of-the-art solutions. The proposed techniques include several *design time* approaches to address aging of the most critical microprocessor components. Besides, also a *dynamic runtime adaptation* scheme was developed to detect and avoid potentially critical system conditions while the system is running. This solution complements the design

[1] Cross-Layer means that the knowledge and parameters available at multiple abstraction layers are used in combination to optimize the design, whereas single-layer solutions exploit only the information of a single level.

Workshop on Early Reliability Modeling for Aging and Variability in Silicon Systems – March 18th 2016 – Co-Located with DATE 2016 - Dresden, Germany

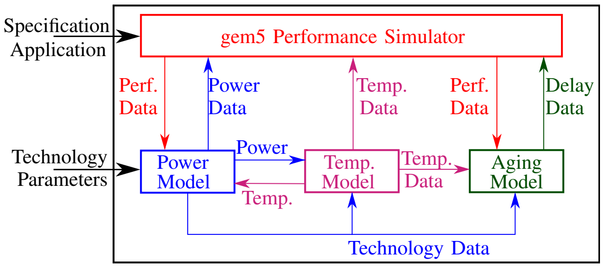


Fig. 4. ExtraTime framework for aging modeling and evaluation

time techniques, which are incapable of dealing with runtime variations (e.g. changing system conditions). Because of that, the design time solutions consider lifetime as one optimization objective and tune the design accordingly based on a given set of representative workload scenarios, while the runtime technique deals with the constantly changing conditions and adapts the system to avoid critical states. Thus, the combination of both schemes enables effective and holistic aging mitigation solutions, which allow a very aggressive and cost-effective system design.

In the following, the different contributions are explained in detail.

II. Frameworks for Wearout Modeling and Evaluating

The first framework developed, is an architectural platform called *ExtraTime* [14]. It is based on the performance simulator gem5 [15] which was extended with sophisticated models for power and temperature. In order to make these models as realistic as possible, they were optimized, and afterwards calibrated and validated using a real experimental platform based on recent Intel Core-i-processors, which have on-die power and thermal sensors [16]. As a result, the model accuracy is very good (e.g. temperature inaccuracy is $< 2°\mathrm{C}$). In addition, novel and realistic architectural aging models were developed and incorporated (see Fig. 4). The main advantage of these models is that they do not require detailed circuit-level information to estimate the degradation of a complete architectural component (e.g. ALU, Branch Predictor, etc.). For this purpose, these models were derived from transistor-level models for BTI and HCI [17]–[19] by introducing a representative transistor, which reflects the average usage behavior (switching activity, ON time, OFF time) of all transistors within this block. In addition, the temperature of this representative transistor is estimated by the block temperature. By that means, the degradation of the representative transistor can be obtained, and thus the degradation of the entire block can be estimated. In this regard it is important to note that the accuracy of the resulting aging models is very good given their level of abstraction. In fact, the inaccuracy compared to accurate gate-level models for an architectural block such as an ALU is less than 5 % without requiring detailed circuit–level knowledge.

As a result, this framework considers the influence of parameters at microarchitecture- up to application-level. Moreover, as ExtraTime does not require low-level details (e.g. the actual gate-level implementation), it can be employed in early design phases for a first-order aging analysis and design space exploration.

To also take low-level aspects into account, a second complementary platform was developed, which is based on standard EDA tools for design synthesis and simulation. It can analyze all internal gates, and it considers the interplay of real-world applications, aging, power and temperature [20]. Thus, it is very accurate as aging can be analyzed at gate-level using the models proposed in [17]–[19], but is less flexible compared to ExtraTime. Consequently, it is intended for fine tuning in later design phases.

III. Design Time & Runtime Aging Mitigation Solutions

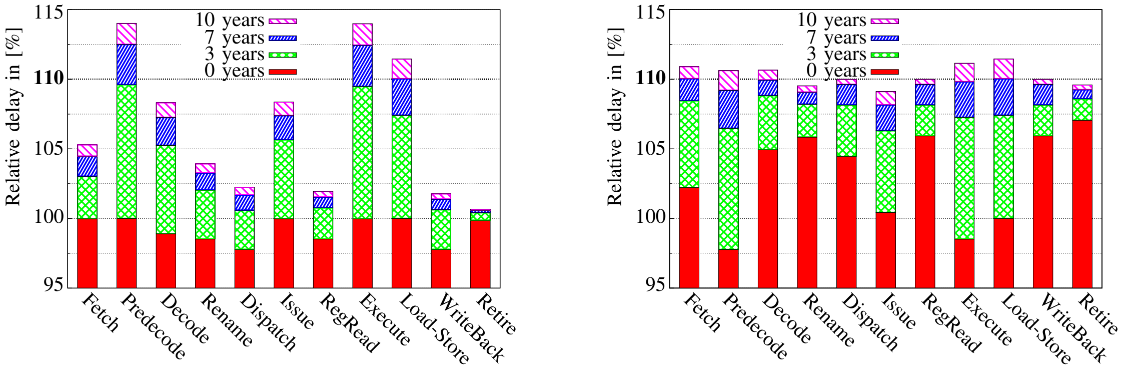
Using these novel cross-layer platforms the most critical microprocessor components were identified and several unique aging mitigation techniques were designed and evaluated, which are presented in the following subsections.

A. Aging-Aware Design of Instruction Pipelines

Traditionally the delays of all instruction pipeline stages are balanced at design time. However, transistor aging causes a non-uniform delay degradation among all stages due to different usage patterns (see Fig. 5(a)). Hence, this design approach results in an imbalanced and non-optimal design after a short period of time. Consequently, a single stage becomes the bottleneck for the overall processor lifetime. In other words, while one pipeline stage already produces timing failures, the other stages still operate correctly. To alleviate this problem, a novel instruction pipeline design paradigm is proposed (*MTTF-balanced pipeline*) according to which all stage delays are balanced at the end of the desired lifetime (see Fig. 5(b)). The main idea of this approach is to increase the timing slack of aging-critical stages to improve their lifetime, while the timing slack of non-critical stages can be reduced to improve their energy efficiency by using slower yet more energy efficient gates. As a result, the processor lifetime can be considerably improved by more than 2.3×, and at the same time the power consumption can be reduced by 10 %. In addition, performance and cost are not affected [20]. This underlines, that it is much more effective to address aging already in early design phases, rather than only adding guardbands to the final design to cope with the delay degradation.

B. Aging-Aware Cross-Layer Instruction Scheduling

As shown in Fig. 5(a), the execution units belong to the most aging-critical processor components. Therefore, an aging-aware instruction scheduling technique was developed [21]. The novelty of this scheduling policy is that the *timing-criticality* of instructions (see Fig. 6) is considered during the scheduling process to reduce the load of units that execute critical instructions. Therefore, the



(a) Delay-Balanced Design

(b) MTTF-Balanced Design

Fig. 5. Delay degradation of the delay-balanced design and MTTF-balanced design for the FabScalar microprocessor [13]

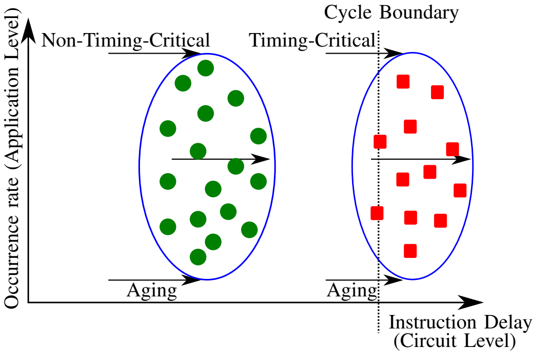


Fig. 6. Illustration of the timing criticality of instructions supported by a functional unit (e.g. ALU)

circuit-level delay of all instructions as well as their application-level occurrence rates were analyzed to classify the instructions into timing-critical (those that start to fail first) and non-timing-critical instructions. Then, dedicated functional units are used for the different instruction classes. Since less than 20 % of all executed instructions are timing-critical, the unit(s) taking care of these instructions are idle most of the time, which is exploited to considerably improve the overall lifetime of the functional units[2] by employing input vector control or aggressive power gating policies. In fact, our simulation results obtained with ExtraTime show that the overall lifetime can be improved by more than 1.6× compared to existing scheduling policies that ignore the detailed timing information (i.e. these are single-layer solutions) and instead balance the number of incoming instructions among all available units.

C. Aging-Aware Instruction Set Encoding

Beside the execution units also the decoding stages of a microprocessor can become critical and limit the microprocessor lifetime (see Fig. 5(a)). Hence, the decoding stages require an aging-aware design. Since the instruction set encoding, i.e. the mapping between instructions and opcodes, has a strong influence on the wearout of the decoding stages (see Fig. 7), we propose a novel aging-aware instruction set encoding methodology called *ArISE* to address the delay degradation in the decoding stages [22]. This methodology exploits simulated annealing and genetic algorithms to optimize the instruction set encoding with respect to lifetime as well as power consumption, since exhaustive optimization solutions are infeasible due to the large number of possible encodings (i.e. typically more than 10^{200}). The result is an optimization that yields significant lifetime improvements (more than 2× compared to state-of-the-art) with negligible impact on other design parameters. This is due to the fact that power consumption and lifetime are co-optimized in our proposed approach which iteratively improves the encoding. If only one of these two parameters is considered, there will be considerable disadvantages for the other one as pointed out in [22].

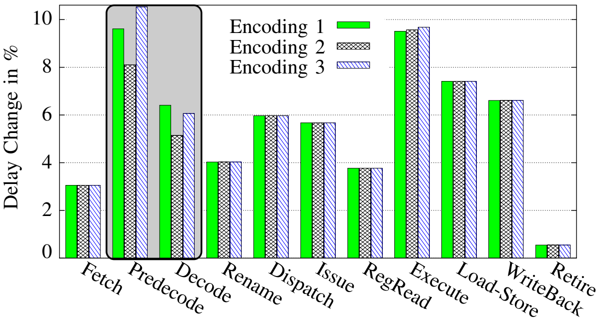


Fig. 7. Aging after 3 years for the FabScalar microprocessor [13] for different encodings

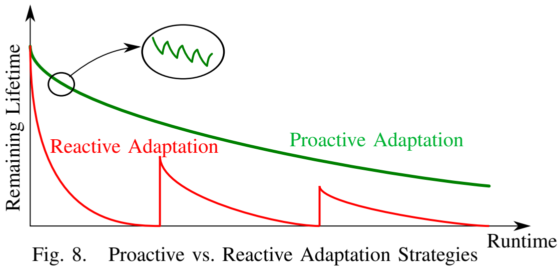


Fig. 8. Proactive vs. Reactive Adaptation Strategies

D. Pro-Active Aging-Aware Dynamic Runtime Adaptation

To detect and avoid potentially critical conditions while the system is running, dynamic schemes employed at runtime have to complement solutions applied at design time [23]. However, the dynamic state-of-the-art techniques employ only reactive adaptation techniques. These are inefficient due to the nature of “damage control”-type of policies, i.e., they deal with already “aged” chips. In contrast, we propose a *proactive* and *preventive* runtime adaptation policy that tries to proactively slow down aging in all phases of the chip lifetime, and hence can prolong the lifetime more efficiently than the existing techniques (see Fig. 8), i.e. with lower performance and power overheads [24]. Therefore, an hierarchical expert system was developed (see Fig. 9) that takes input from a sensor network (or models running in software) to analyze the current system state as well as the trend of recent system states in a very fine-grained manner, i.e. every 1 ms-10 ms and adapts the system accordingly. Whenever a critical state or trend in terms of lifetime, temperature or power consumption is detected, the system is adapted by means of frequency and voltage tuning, that is, frequency and voltage are reduced by one level. If no parameter as well as no trend is critical, the current system performance is evaluated. If the performance can be maintained with a lower frequency level, frequency and supply voltage are lowered to improve lifetime, power consumption and temperature, otherwise the frequency is kept on the same level or is even increased if necessary. As a result, the lifetime of the entire microprocessor can be improved by more than 2×, and the energy consumption can be reduced by 14 %, while the performance penalty is almost negligible (2 % on average). This shows that with such a cross-layer, proactive runtime adaptation technique the different design parameters can be co-optimized very effectively although the adaptation decisions are performed at system-level.

IV. Summary

In this work cross-layer solutions for aging modeling and simulation as well as mitigation were pushed forward. Therefore, a set of unique frameworks and mitigation techniques were developed. In addition, it was demonstrated that cross-layer solutions allow a much more efficient co-optimization of all design parameters including lifetime compared to state-of-the-art single-layer solutions.

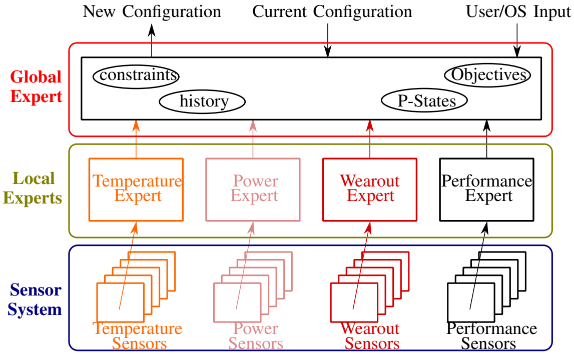


Fig. 9. Organization of the expert system

[2] Please note that still the units taking care of the timing-critical instructions limit the overall lifetime due to the way the instructions are classified.

V. Acknowledgement

This work was partly supported by the German Research Foundation (DFG) as part of the national focal program "Dependable Embedded Systems" (SPP-1500, http://spp1500.ira.uka.de).

VI. References

[1] T. Mak, "Is CMOS More Reliable with Scaling?" in *Proceedings of the Online Testing Workshop*, Jul. 2002.

[2] H. Nguyen, "Resiliency Challenges in Future Communications Infrastructure," in *Proceedings of the Communications Quality and Reliability Workshop*, May 2014.

[3] International Technology Roadmap for Semiconductors, in *ITRS 2013 Edition – Process Integration, Devices, and Structures*, 2014.

[4] S. Borkar *et al.*, "The Future of Microprocessors," *Communications of the ACM*, pp. 67–77, May 2011.

[5] S. Mitra *et al.*, "Robust System Design to Overcome CMOS Reliability Challenges," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, pp. 30–41, Mar. 2011.

[6] S. Borkar, "Designing Reliable Systems from Unreliable Components: The Challenges of Transistor Variability and Degradation," *IEEE Micro*, pp. 10–16, Nov. 2005.

[7] V. Narayanan *et al.*, "Reliability Concerns in Embedded System Designs," *Computer*, pp. 118–120, Jan. 2006.

[8] M. Agarwal *et al.*, "Circuit Failure Prediction and Its Application to Transistor Aging," in *Proceedings of the VLSI Test Symposium*, May 2007, pp. 277–286.

[9] J. Henkel *et al.*, "Design and architectures for dependable embedded systems," in *Proceedings of the Conference on Hardware/Software Codesign and System Synthesis*, Oct. 2011, pp. 69–78.

[10] European Nanoelectronics Initiative Advisory Council, *ENIAC Strategic Research Agenda - European Technology Platform Nanoelectronics*, 2nd ed. ENIAC, 2007.

[11] *Bias Temperature Instability in HKMG MOSFETs: Characterization, Process Dependence, DC/AC Modeling and Stochastic Effects*, Mar. 2014, Tutorial.

[12] X. Li *et al.*, "Compact Modeling of MOSFET Wearout Mechanisms for Circuit-Reliability Simulation," *IEEE Transactions on Device and Materials Reliability*, pp. 98–121, Mar. 2008.

[13] N. Choudhary *et al.*, "FabScalar: Automating Superscalar Core Design," *IEEE Micro*, pp. 48–59, May 2012.

[14] F. Oboril *et al.*, "ExtraTime: Modeling and Analysis of Wearout due to Transistor Aging at Microarchitecture-Level," in *Proceedings of the International Conference on Dependable Systems and Networks*, Jun. 2012, pp. 1–12.

[15] N. L. Binkert *et al.*, "The M5 Simulator: Modeling Networked Systems," *IEEE Micro*, pp. 52–60, Jul. 2006.

[16] F. Oboril *et al.*, "High-Resolution Online Power Monitoring for Modern Microprocessors," in *Proceedings of the Conference on Design, Automation and Test in Europe*, Mar. 2014, pp. 1–4, to appear.

[17] W. Wang *et al.*, "Compact Modeling and Simulation of Circuit Reliability for 65-nm CMOS Technology," *IEEE Transactions on Device and Materials Reliability*, pp. 509–517, Dec. 2007.

[18] E. Takeda *et al.*, "New hot-carrier injection and device degradation in submicron MOSFETs," *IEEE Proceedings I, Solid-State and Electron Devices*, pp. 144–150, Jun. 1983.

[19] A. Bravaix *et al.*, "Hot-Carrier Acceleration Factors for Low Power Management in DC-AC stressed 40nm NMOS node at High Temperature," in *Proceedings of the International Reliability Physics Symposium*, Apr. 2009, pp. 531–548.

[20] F. Oboril *et al.*, "Aging-Aware Design of Microprocessor Instruction Pipelines," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 704–716, May 2014.

[21] F. Oboril *et al.*, "Negative Bias Temperature Instability-Aware Instruction Scheduling: A Cross-Layer Approach," *Journal of Low Power Electronics*, pp. 389–402, Dec. 2013.

[22] F. Oboril *et al.*, "Exploiting Instruction Set Encoding for Aging-Aware Microprocessor Design," *ACM Transactions on Design Automation of Electronic Systems*, pp. 1–23, 2015, to appear.

[23] P. Gupta *et al.*, "Underdesigned and Opportunistic Computing in Presence of Hardware Variability," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 8–23, Jan. 2013.

[24] F. Oboril *et al.*, "Reducing Wearout in Embedded Processors using Proactive Fine-Grained Dynamic Runtime Adaptation," in *Proceedings of the European Test Symposium*, May 2012, pp. 68–73.