Report from Dagstuhl Seminar 12341

Verifying Reliability

Edited by

Görschwin Fey[1], Masahiro Fujita[2], Natasa Miskov-Zivanov[3], Kaushik Roy[4], and Matteo Sonza Reorda[5]

1 **German Aerospace Center (DLR) and University of Bremen, DE,** fey@informatik.uni-bremen.de
2 **University of Tokyo, JP,** fujita@ee.t.u-tokyo.ac.jp
3 **University of Pittsburgh, US,** nam66@pitt.edu
4 **Purdue University, US,** kaushik@ecn.purdue.edu
5 **Politecnico di Torino, IT,** matteo.sonzareorda@polito.it

Abstract

Moore's law has been the driving force behind the increasing computing power of today's devices which is based on shrinking feature sizes. This shrinking process makes future devices extremely susceptible to soft errors due to, e.g., external influences like environmental radiation and internal issues like stress effects, aging and process variation. For future technology nodes "Designing reliable systems from unreliable components" [1] will be one of the most important topics.

Seminar 19.–24. August, 2012 – http://www.dagstuhl.de/12341
1998 ACM Subject Classification B.6.2 Reliability and Testing [**] (B.8); B.6.3 Design Aids; B.7.3 Reliability and Testing [**] (B.8); B.7.2 Design Aids
Keywords and phrases Reliability, fault modeling, formal methods
Digital Object Identifier 10.4230/DagRep.2.8.54
Edited in cooperation with Mehdi Dehbashi (University of Bremen, DE)

1 Executive Summary

Görschwin Fey
Masahiro Fujita
Natasa Miskov-Zivanov
Kaushik Roy
Matteo Sonza Reorda

License Creative Commons BY-NC-ND 3.0 Unported license
© Görschwin Fey, Masahiro Fujita, Natasa Miskov-Zivanov, Kaushik Roy, and Matteo Sonza Reorda

Introduction

Moore's law predicted the ever increasing computing power of the past decades from an economic perspective based on doubling the number of elements in a circuit about every two years. Moreover, Moore's law is expected to continue for another 10-20 years. On the physical level this integration is enabled by continuously shrinking feature sizes of basic components. But for future technology nodes reliability problems triggered by different

[1] Shekhar Y. Borkar, "Designing reliable systems from unreliable components: the challenges of transistor variability and degradation," IEEE Micro 25(6): 10-16 (2005)

Except where otherwise noted, content of this report is licensed under a Creative Commons BY-NC-ND 3.0 Unported license

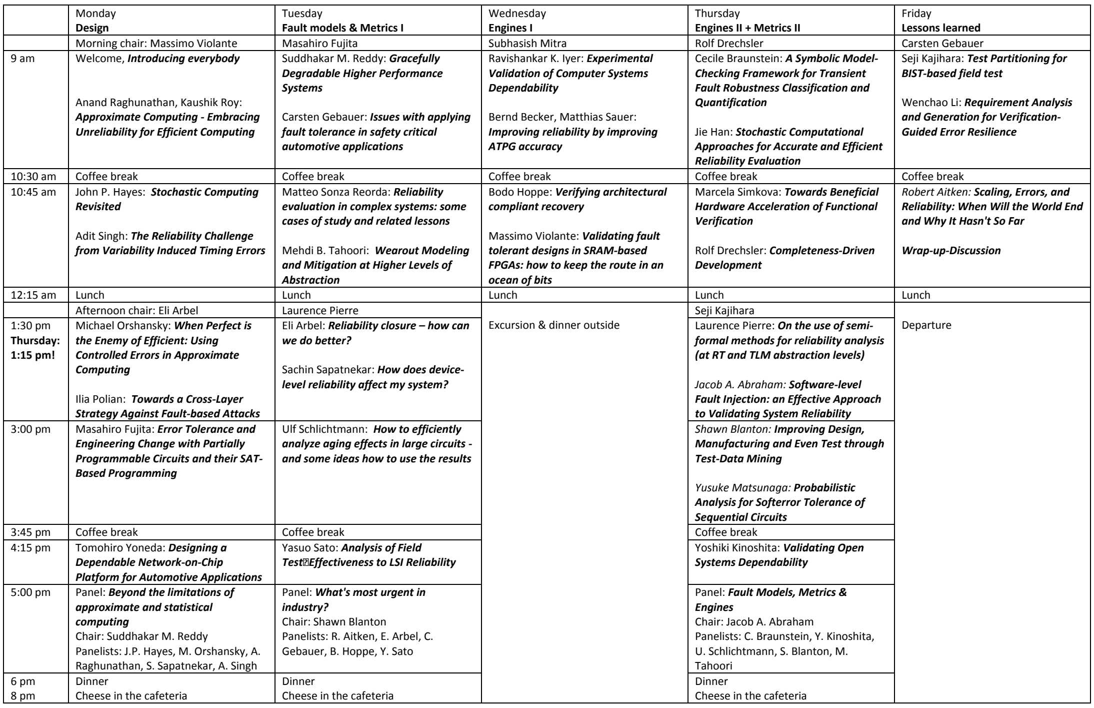
Verifying Reliability, *Dagstuhl Reports*, Vol. 2, Issue 8, pp. 54–73
Editors: Görschwin Fey, Masahiro Fujita, Natasa Miskov-Zivanov, Kaushik Roy, and Matteo Sonza Reorda
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

sources are expected to increase rapidly. Process variations in the production process are one issue. While production processes become more accurate considering absolute measures, the relative inaccuracy compared to the component's size is increasing. One consequence are transistors with a wide range of threshold levels resulting in slightly faster or slower operating logic circuitry (both die-to die and within die). This may result for example in delay errors under certain operating conditions of a device. Increasing sensitivity to the omnipresent environmental radiation is another issue. In the past some errors induced by radiation have been observed infrequently while systems in space missions are already specified to be radiation resistant. Shrinking feature sizes result in sensitivity to radiation with lower energy causing more radiation induced events like Single Event Upsets (SEUs) even on sea level. Such effects are summarized as transient faults resulting in soft errors (as opposed to permanent faults resulting in a change of the functionality due to a modification of the physical structure). Consequently, approaches to design reliable circuits tolerating such transient faults without causing soft errors have been proposed. These design approaches to mitigate soft errors comprise all levels of design abstraction from the system specification down to the layout. Examples for these approaches are, e.g., fault tolerant algorithms and operating systems, fault tolerant processors, self-calibrating architectures, block level redundancy and error checking, synthesis approaches on the gate level, or hardening techniques on the layout level. In practical systems typically multiple mitigation techniques are implemented to guarantee reliability across the full system stack. Functional verification has been and still is a challenge in current designs containing up to hundreds millions of transistors. Mature techniques for the formal verification and the dynamic verification of large systems exist. Research in verification is ongoing to match the rapid increase of the size of the systems. The verification of reliability is an interdisciplinary topic involving at least testing technology, verification methodology, and design experience. This makes the verification of reliable implementations an even harder problem. The testing community provides underlying models for transient faults to understand the effects at the functional and eventually at the system level. Using these models, the verification community designs efficient analysis tools and verification techniques to handle large systems. As in standard verification of large circuits a concerted action of formal methods, semi-formal techniques and simulation-based validation will be required. Still knowledge from the design community is required, to further speed up the verification task. Understanding the implemented approach to reliability on the application level and the system level is required to achieve a high degree of automation in the verification task.

Organization

The seminar was organized in short slots for talks followed by extensive discussions. A panel discussion in the afternoon summarized each day and focused on further questions (Figure 1). Each day was devoted to a special topic:

- Design – Techniques to ensure reliability by design.
- Fault models – Different types of fault models are required depending on the abstraction level and the type of design considered.
- Metrics – Measuring reliability requires some kinds of metrics. These metrics can be defined with respect to the fault models. But they should also reflect potential inaccuracies.
- Engines – Different types of engines are used in Electronic Design Automation (EDA) for circuits and systems.



	Monday **Design**	Tuesday **Fault models & Metrics I**	Wednesday **Engines I**	Thursday **Engines II + Metrics II**	Friday **Lessons learned**
	Morning chair: Massimo Violante	Masahiro Fujita	Subhasish Mitra	Rolf Drechsler	Carsten Gebauer
9 am	Welcome, ***Introducing everybody*** Anand Raghunathan, Kaushik Roy: ***Approximate Computing - Embracing Unreliability for Efficient Computing***	Suddhakar M. Reddy: ***Gracefully Degradable Higher Performance Systems*** Carsten Gebauer: ***Issues with applying fault tolerance in safety critical automotive applications***	Ravishankar K. Iyer: ***Experimental Validation of Computer Systems Dependability*** Bernd Becker, Matthias Sauer: ***Improving reliability by improving ATPG accuracy***	Cecile Braunstein: ***A Symbolic Model-Checking Framework for Transient Fault Robustness Classification and Quantification*** Jie Han: ***Stochastic Computational Approaches for Accurate and Efficient Reliability Evaluation***	Seji Kajihara: ***Test Partitioning for BIST-based field test*** Wenchao Li: ***Requirement Analysis and Generation for Verification-Guided Error Resilience***
10:30 am	Coffee break	Coffee break	Coffee break	Coffee break	Coffee break
10:45 am	John P. Hayes: ***Stochastic Computing Revisited*** Adit Singh: ***The Reliability Challenge from Variability Induced Timing Errors***	Matteo Sonza Reorda: ***Reliability evaluation in complex systems: some cases of study and related lessons*** Mehdi B. Tahoori: ***Wearout Modeling and Mitigation at Higher Levels of Abstraction***	Bodo Hoppe: ***Verifying architectural compliant recovery*** Massimo Violante: ***Validating fault tolerant designs in SRAM-based FPGAs: how to keep the route in an ocean of bits***	Marcela Simkova: ***Towards Beneficial Hardware Acceleration of Functional Verification*** Rolf Drechsler: ***Completeness-Driven Development***	*Robert Aitken:* ***Scaling, Errors, and Reliability: When Will the World End and Why It Hasn't So Far*** ***Wrap-up-Discussion***
12:15 am	Lunch	Lunch	Lunch	Lunch	Lunch
	Afternoon chair: Eli Arbel	Laurence Pierre	Excursion & dinner outside	Seji Kajihara	Departure
1:30 pm **Thursday: 1:15 pm!**	Michael Orshansky: ***When Perfect is the Enemy of Efficient: Using Controlled Errors in Approximate Computing*** Ilia Polian: ***Towards a Cross-Layer Strategy Against Fault-based Attacks***	Eli Arbel: ***Reliability closure – how can we do better?*** Sachin Sapatnekar: ***How does device-level reliability affect my system?***		Laurence Pierre: ***On the use of semi-formal methods for reliability analysis (at RT and TLM abstraction levels)*** *Jacob A. Abraham:* ***Software-level Fault Injection: an Effective Approach to Validating System Reliability***	
3:00 pm	Masahiro Fujita: ***Error Tolerance and Engineering Change with Partially Programmable Circuits and their SAT-Based Programming***	Ulf Schlichtmann: ***How to efficiently analyze aging effects in large circuits - and some ideas how to use the results***		*Shawn Blanton:* ***Improving Design, Manufacturing and Even Test through Test-Data Mining*** *Yusuke Matsunaga:* ***Probabilistic Analysis for Softerror Tolerance of Sequential Circuits***	
3:45 pm	Coffee break	Coffee break		Coffee break	
4:15 pm	Tomohiro Yoneda: ***Designing a Dependable Network-on-Chip Platform for Automotive Applications***	Yasuo Sato: ***Analysis of Field Test�Effectiveness to LSI Reliability***		Yoshiki Kinoshita: ***Validating Open Systems Dependability***	
5:00 pm	Panel: ***Beyond the limitations of approximate and statistical computing*** Chair: Suddhakar M. Reddy Panelists: J.P. Hayes, M. Orshansky, A. Raghunathan, S. Sapatnekar, A. Singh	Panel: ***What's most urgent in industry?*** Chair: Shawn Blanton Panelists: R. Aitken, E. Arbel, C. Gebauer, B. Hoppe, Y. Sato		Panel: ***Fault Models, Metrics & Engines*** Chair: Jacob A. Abraham Panelists: C. Braunstein, Y. Kinoshita, U. Schlichtmann, S. Blanton, M. Tahoori	
6 pm 8 pm	Dinner Cheese in the cafeteria	Dinner Cheese in the cafeteria		Dinner Cheese in the cafeteria	

Figure 1 Seminar schedule

Results

Documenting the results of intensive discussions in a compact manner is difficult. However, some results can be formulated in crisp statements. Approximate computing is a powerful technique for reliable design where the applications permit inaccuracy of operations up to a certain extent. Computing considering statistical nature of devices may be able to produce very accurate results, but providing compatible computing fabric at acceptable costs is a challenge. No single fault model will cover all aspects of reliability. In particular, fault models must be adapted to the application domain, the level of criticality and the step in the design process that is being considered. Appropriate metrics will then be applied to bridge gaps, e.g., between different levels of abstraction. An orchestration of reasoning engines ranging from formal techniques to simulation and emulation will always be required to gather data required for the different metrics. Design for Reliability will always affect all levels of abstraction. Only by concerted effort the same performance gains can be expected that we have seen in the past 50 years.

As a follow-up of the Dagstuhl Seminar, an Embedded Tutorial was successfully proposed for the DATE conference 2013. The Embedded Tutorial's title is "Reliability Analysis Reloaded: How Will We Survive?" and will include two presentations given by participants of the seminar or colleagues belonging to the research group of a participant.

2 Table of Contents

Executive Summary

Overview of Talks

Panel Discussions

3 Overview of Talks

3.1 Software-Level Fault Injection: An Effective Approach to Validating System Reliability

Jacob A. Abraham (Univ. of Texas at Austin, US)

License 🅭 🅮 ⊜ Creative Commons BY-NC-ND 3.0 Unported license
© Jacob A. Abraham

Accurate evaluation of the reliability of a complex system is extremely difficult since faults at the hardware level have to be analyzed with respect to their impact on the system under varying operating conditions and workloads. Simulating a system for billions of processor cycles for different types of low-level faults is practically impossible. This talk will describe techniques which evaluate the dependability of a system by running it under normal conditions and using software routines to inject faults which emulate the effect of the hardware on system behavior. Examples of applying the ideas to a variety of systems will be described, as well as directions for exploiting virtualization and other hardware support provided by modern processors.

3.2 Scaling, Errors, and Reliability: When Will the World End and Why It Hasn't So Far

Robert Aitken (ARM Inc. - San Jose, US)

License 🅭 🅮 ⊜ Creative Commons BY-NC-ND 3.0 Unported license
© Robert Aitken

Technology scaling continues to follow the basics of Moore's Law, despite difficult challenges in lithography, materials, and design. Looking at what has succeeded so far can give insight into why several predicted demises of scaling have not happened, as well as showing which of the current candidates might actually succeed. Gordon Moore said "No exponential is forever, but we can delay forever" - when will forever arrive?

3.3 Reliability Closure – How Can We Do Better?

Eli Arbel (IBM - Haifa, IL)

License 🅭 🅮 ⊜ Creative Commons BY-NC-ND 3.0 Unported license
© Eli Arbel

A typical design process involves dealing with multiple constraints, such as power, performance and timing. As reliability is becoming increasingly important in many applications, it can be viewed as an additional design constraint which should be met before closing the design. Many reliability features are implemented in the RT-level, thus it is of high importance to provide feedback to logic designers whether their design meets its reliability goals, and if not assist them with rectifying reliability issues. We will present some techniques for analyzing design vulnerability to soft-errors at the RT-level, how they canbe used to facilitate logic implementation with respect to reliability and discuss how we can help design teams achieve their reliability goals in more accurate and faster ways.

3.4 Improving Reliability by Improving ATPG Accuracy

Bernd Becker, Matthias Sauer (Universität Freiburg, DE)

License Creative Commons BY-NC-ND 3.0 Unported license
© Bernd Becker, Matthias Sauer

We present SAT-based approaches - implemented in the tools Phaeton and WaveSAT- for the analysis of circuit timing and the detection of small delay defects. Phaeton enumerates all or a user-specified number of longest sensitisable paths in the whole circuit or through specific components. The algorithm encodes all aspects of the path search as an instance of the Boolean Satisfiability Problem(SAT), which allows the method not only to benefit from recent advances in SAT-solving technology, but also to avoid some of the drawbacks of previous structural approaches. The path information obtained by Phaeton can be used for several applications including design and test of circuits affected by statistical process variations, criticality analysis, and post silicon debug. The approach has been extended to sequential ATPG making use of recent advances in Bounded Model Checking. However, the computation of small-delay fault test patterns by path sensitization may result in false positives and false negatives as well. We developed WaveSAT, a SAT-based automatic test pattern generation algorithm which considers waveforms and their propagation on each relevant line of the circuit. The model incorporates individual delays for each gate and filtering of small glitches. WaveSAT generates a test if the fault is testable and is also capable of automatically generating a formal redundancy proof for undetectable small-delay faults. Experimental results for academic and industrial benchmark circuits demonstrate the methods' accuracy and scalability.

3.5 Improving Design, Manufacturing and Even Test through Test-Data Mining

Shawn Blanton (Carnegie Mellon University - Pittsburgh, US)

License Creative Commons BY-NC-ND 3.0 Unported license
© Shawn Blanton

For many years now, ACTL (Advanced Chip Test Laboratory, www.ece.cmu.edu/ actl) at Carnegie Mellon has been using layout information for improving manufacturing test, in particular, for changing test from a sort-only process to one that also involves learning about the design, the manufacturing process, and their interaction in producing high-yielding, high-quality parts. In this talk, I will describe METER (MEasuring Test Effectiveness Regionally), a novel approach that measures the effectiveness of arbitrary fault models and test metrics through the statistical analysis or readily-available tester data.

3.6 A Symbolic Model-Checking Framework for Transient Fault Robustness Classification and Quantification

Cecile Braunstein (UPMC - Paris, FR)

License 🅭 🅯 🅮 Creative Commons BY-NC-ND 3.0 Unported license
© Cecile Braunstein

Robustness analysis of RTL-sequential circuits impacted by transient faults is an important concern for designers. While simulation or emulation based techniques are widely used, they do not give guarantees on the robustness level of the system and are often limited to single fault models. Moreover, several robustness criterion may be adopted depending on the application being executed and the synchronisation scheme between the circuit and its environment. The use of formal methods ensures robustness level and helps in locating weak portions of a circuit to be hardened, even in case of multiple fault models. We present a framework to analyse the robustness of a RTL circuit, considering several models of faults and reparation, and show how a wideclass of robustness criteria can be mapped into our reparation model. We present an implementation of the robustness measures in the setting of BDD-based model checking and illustrate our measurements on classical benchmark circuits.

3.7 Completeness-Driven Development

Rolf Drechsler (Universität Bremen and DFKI Bremen, DE)

License 🅭 🅯 🅮 Creative Commons BY-NC-ND 3.0 Unported license
© Rolf Drechsler

Due to the steadily increasing complexity, the design of embedded systems faces serious challenges. To meet these challenges additional abstraction levels have been added to the conventional designflow resulting in Electronic System Level (ESL) design. Besides abstraction, the focus in ESL during the development of a system moves from design to verification, i.e. checking whether or not the system works as intended becomes more and more important. However, at each abstraction level only the validity of certain properties is checked. Completeness, i.e. checking whether or not the entire behavior of the design has been verified, is usually not continuously checked. As a result, bugs may befound very late causing expensive iterations across several abstraction levels. This delays the finalization of the embedded system significantly. In this work, we present the concept of Completeness-Driven Development(CDD). Based on suitable completeness measures, CDD ensures that the next step in the design process can only be entered if completeness at the current abstraction level has been achieved. This leads to an early detection of bugs and accelerates the whole design process. The application of CDD is illustrated by means of an example.

3.8 Error Tolerance and Engineering Change with Partially Programmable Circuits and their SAT-Based Programming

Masahiro Fujita (University of Tokyo, JP)

License Creative Commons BY-NC-ND 3.0 Unported license
© Masahiro Fujita

Introducing partial programmability in circuits by replacing some gates with *Look Up Tables* (LUTs) can be an effective way to improve post-silicon or in-field rectification and debugging. Although finding configurations of LUTs that can correct the circuits can be formulated as a QBF problem, solving it by state-of-the-art QBF solvers is still a hard problem for large circuits and many LUTs. In this paper, we present a rectification and debugging method for combinational circuits with LUTs by repeatedly applying Boolean SAT solvers. Through the experimental results, we show our proposed method can quickly find LUT configurations for large circuits with many LUTs, which cannot be solved by a QBF solver.

3.9 Issues with Applying Fault Tolerance in Safety Critical Automotive Applications

Carsten Gebauer (Robert Bosch GmbH - Schwieberdingen, DE)

License Creative Commons BY-NC-ND 3.0 Unported license
© Carsten Gebauer

Fault tolerance will be one of the key features necessary to be able to take advantage from the ever shrinking process technologies. However applying fault tolerance also has its risks. Within this talk I would like to present from an automotive point of view the issues we see regarding safety and ask for possible solutions to address these issues, in particular - latent faults of ISO 26262 - *Error Correction Codes* (ECC): Reduction of error detection due to application of correction - what is reasonable to tolerate, what not?

3.10 Stochastic Computational Approaches for Accurate and Efficient Reliability Evaluation

Jie Han (University of Alberta, CA)

License Creative Commons BY-NC-ND 3.0 Unported license
© Jie Han

Reliability is fast becoming a major concern due to the nanometric scaling of CMOS technology. Accurate analytical approaches for the reliability evaluation of logic circuits, however, have a computational complexity that generally increases exponentially with circuit size. This makes intractable the reliability analysis of large circuits. This talk initially presents novel computational models based on stochastic computation; using these stochastic computational models (SCMs), a simulation-based analytical approach is then proposed for the reliability evaluation of logic circuits. In this approach, signal probabilities are encoded in the statistics of random binary bit streams and non-Bernoulli sequences of random permutations of binary bits are used for initial input and gate error probabilities. By leveraging the bit-wise dependencies of random binary streams, the proposed approach takes into account signal correlations and evaluates the joint reliability of multiple outputs. Therefore, it accurately determines the reliability of a circuit; its precision is only limited by the random fluctuations inherent in the stochastic sequences. Based on both simulation and analysis, the SCM approach takes advantages of ease in implementation and accuracy in evaluation. The use of non-Bernoulli sequences as initial inputs further increases the evaluation efficiency and accuracy compared to the conventional use of Bernoulli sequences, so the proposed stochastic approach is scalable for analyzing large circuits. It can further account for various fault models as well as calculating the soft error rate (SER). These results are supported by extensive simulations and detailed comparison with existing approaches.

3.11 Stochastic Computing Revisited

John P. Hayes (University of Michigan, US)

License Creative Commons BY-NC-ND 3.0 Unported license
© John P. Hayes

Stochastic computing (SC) was proposed in the 1960s as a low-cost alternative to conventional computing with weighted binary numbers. It is unusual in that it represents information by means of long pseudo-random bit-streams, which can be interpreted as probabilities and processed by low-cost standard logic circuits. The SC approach is well-suited to some important new applications that require massive parallelism or extremely high tolerance of soft errors, such as ECC controllers for WiFi and flash memories. Despite some success in specialized application areas, many aspects of SC are poorly understood and a comprehensive design methodology has yet to emerge. This talk will review SC and its status from a modern perspective, focusing on design and verification issues affecting accuracy, area cost, and reliability. A novel approach to SC circuit design based on spectral transforms will also be presented.

3.12 Verifying Architectural Compliant Recovery

Bodo Hoppe (IBM Deutschland - Böblingen, DE)

License Creative Commons BY-NC-ND 3.0 Unported license
© Bodo Hoppe

A lot of research has been done in investing whether soft errors can be detected in hardware designs. Structural and static analysis are used to ensure any error can be detected. Formal analysis can be used to verify detectability of errors. However, today exhaustive verification has to be executed to verify that the design is reliable and able to recover properly according to the architecture. This has a lot of design complexities especially in presence of a multicore environment with coherent memory as well as high-frequency supüerscalarmicroprocessor designs with hardware recovery functionality. A lot of side effects may occur in the hardware and may lead to unwanted effects, for example*Potential Unexpected Loss of Data*(PULD). A method is being presented, that improves significantly the efficiency of proving that the design actually can maintain the architectural state including non-corrupted memory. But a much bigger question is can a certain design approach or rules can allow an easyp roof that exhaustive fault simulation can be avoided. Or can design rulechecker in a combination with formal verification be used to ensure recoverability of the design?

3.13 Test Partitioning for BIST-Based Field Test

Seiji Kajihara (Kyushu Institute of Technology, JP)

License Creative Commons BY-NC-ND 3.0 Unported license
© Seiji Kajihara

A BIST-based field test has been used to guarantee high reliability of VLSIs. But it is not easy for field test to achieve high test quality due to the limitation of short test application time. Test partitioning and rotating test is an effective way to satisfy such a constraint. A test set of a circuit is partitioned into a number of subsets, and apply only one subset to the circuit at one test session of field test. On the other hand, because test partitioning would cause fault coverage loss at each test session, aging-induced faults will be undetected at some test sessions. The longer the detection interval is, the higher the likelihood of a system failure would be. In this talk we discuss on a metric to estimate the failure rate for test partitioning and approaches to partition a given test set into several subsets aiming to minimize the failure rate.

3.14 Validating Open Systems Dependability

Yoshiki Kinoshita (AIST - Hyogo, JP)

License Creative Commons BY-NC-ND 3.0 Unported license
© Yoshiki Kinoshita

I wish to convey the idea that checking the appropriateness of the claims to be verified (i.e., validation) is important in achievement of Open Systems Dependability. I also present assurance cases, which documents the result of validation. I shall use the system *European AIS [Aeronautical Information Service] Database* (EAD) as my leading example. Time permitting, I also introduce our ongoing work on D-Case/Agda, a system that supports development and checking of assurance cases using Proof Assistant technology. Verification is critical, but it should be associated with validation. Open Systems Dependability is dependability, the notion evolved from reliability, treated from Open Systems View, which our DEOS project has introduced to consider ever-changing and vague aspects of huge and complicated systems. Many systems have open systems aspects, and hardware seems no exception. Development and maintenance of the assurance case is central in achievement of Open Systems Dependability. Assurance cases are documents where all information about verification and validation of the system is available; they may be considered as "qualitative metric" of dependability.

3.15 Requirement Analysis and Generation for Verification-Guided Error Resilience

Wenchao Li (University of California - Berkeley, US)

License Creative Commons BY-NC-ND 3.0 Unported license
© Wenchao Li

All error resilience techniques employ some form of redundancy, resulting in added cost such as area or power overhead. Formal Verification has been shown to be effective in judiciously guiding the deployment of added error resilience. However, such analysis is sensitive to the quality and completeness of specifications. In this talk, I will discuss works on requirement analysis and generation with application to error resilience, as well as other useful applications such as error localization.

3.16 Probabilistic Analysis for Soft-Error Tolerance of Sequential Circuits

Yusuke Matsunaga (Kyushu University, JP)

License 🅭 🅮 ⊜ Creative Commons BY-NC-ND 3.0 Unported license
© Yusuke Matsunaga

This talk presents a method for estimating the error propagation probabilities in sequential circuits when one ore more FFs' values are changed due to Soft Error Effects.

3.17 When Perfect is the Enemy of Efficient: Using Controlled Errors in Approximate Computing

Michael Orshansky (Univ. of Texas at Austin, US)

License 🅭 🅮 ⊜ Creative Commons BY-NC-ND 3.0 Unported license
© Michael Orshansky

This talk describes our recent work on enabling low-level approximate computation and development of design principles for energy-optimal approximate ("sloppy") addition. We identify a fundamental trade-off between error frequency and error magnitude in a timing-starved adder and introduce a formal model to prove that for signal processing applications using a quadratic signal-to-noise ratio error measure, reducing bit-wise error frequency is sub-optimal. Instead, energy-optimal approximate addition requires limiting maximum error magnitude. The remaining approximation error can be reduced by conditional bounding logic for lower significance bits. We also show how the existence of an intrinsic notion of quality floor present in typical digital signal processing circuits can be used to reduce their energy consumption by strategically accepting some runtime errors. The basic philosophy is to prevent signal quality from severe degradation by using data statistics. The introduced innovations include techniques for carefully controlling possible errors and exploiting the specific patterns of errors for low-cost post-processing to minimize image quality degradation.

3.18 On the Use of Semi-Formal Methods for Reliability Analysis (at RT and TLM Abstraction Levels)

Laurence Pierre (TIMA - Grenoble, FR)

License 🅭 🅮 ⊜ Creative Commons BY-NC-ND 3.0 Unported license
© Laurence Pierre

Evaluating the robustness of hardware systems (ranging from digital IP blocks to complex systems on chip) with respect to soft errors has become an important part of the design flow for many applications, at various levels of abstraction. To avoid the well-known state explosion problem that may occur when using formal approaches (static analysis), we investigate the use of semi-formal (simulationor emulation-based) techniques to improve dependability analysis. We target the analysis of the consequences of soft errors with respect to the application, no matter their origin. We propose to formalize robustness or reliability properties as PSL assertions, and to verify them at runtime using automatically derived property checkers. We discuss two illustrative examples: dependability properties of a cryptographic component for a programmable hardware device to be integrated in networking infrastructures (VHDL RTL description), and safety requirements for an avionics flight control remote module (SystemC TLM description).

3.19 Towards a Cross-Layer Strategy Against Fault-based Attacks

Ilia Polian (Universität Passau, DE)

License Creative Commons BY-NC-ND 3.0 Unported license
© Ilia Polian

Mobile and embedded systems process sensitive information, including personal data such as health records or financial transactions, and confidential parameters of technical systems, e.g., car engines. Protection is provided by cryptographic hardware blocks that are vulnerable to fault-based attacks. Over 700 such attacks have been published so far. Individual, attack-specific countermeasures no longer sufficient. There is strong need for a generic methodology to counter fault-based attacks with reasonable costs. The presentation will introduce fault-based attacks using the recent attack on the LED block ciphers as an example. After that, initial first results on across-layer protection strategy combining specially designed error-detecting codes with selective hardening of individual circuit elements will be presented.

3.20 Approximate Computing - Embracing Unreliability for Efficient Computing

Anand Raghunathan, Kaushik Roy (Purdue University, US)

License Creative Commons BY-NC-ND 3.0 Unported license
© Anand Raghunathan, Kaushik Roy

Designers of computing systems have been remarkably consistent in their approach to unreliability - attempt to eliminate it at all costs. While this approach has its merits and is necessary in some applications, it incurs very high costs in terms of efficiency (power, performance, or cost). With the explosion of digital data, computing platforms are increasingly being used to execute applications (such as web search, data analytics, sensor data processing, recognition, mining, and synthesis) that are inherently resilient or forgiving to errors in most of the computations. This forgiving nature is due to several factors including the redundancy and noisiness of the input data, the statistical nature of the computations themselves, and the acceptability (and often, inevitability) of less-than-perfect results. Approximate computing is an approach to designing computing platforms that are more efficient, by leveraging the forgiving nature of applications. I will outline a range of approximate computing techniques that we have developed from software to architecture to circuits, which have shown promising results. In the context of the increasing unreliability of scaled semiconductor technologies, approximate computing suggests that embracing unreliability rather than attempting to eliminate it could be a promising approach to eschew the high costs of defect and fault tolerance. I will outline some of the challenges that need to be addressed to realize this promise, including a shift in designer mindset, and the development of systematic design methodologies to ensure that unreliability is exposed to applications in a controlled manner.

3.21 Gracefully Degradable Higher Performance Systems

Sudhakar M. Reddy (University of Iowa - Iowa City, US)

License Creative Commons BY-NC-ND 3.0 Unported license
© Sudhakar M. Reddy

In this talk design of systems that permit trade off between performance and defect tolerance is proposed. It is suggested that additional features for defect tolerance should also facilitate enhanced performance when defects are not present or fewer than planned for number of defects are present. Enhanced performance could be higher frequency of operation or higher computational power or additional functionality.

3.22 How does Device-Level Reliability Affect my System?

Sachin Sapatnekar (University of Minnesota, US)

License Creative Commons BY-NC-ND 3.0 Unported license
© Sachin Sapatnekar

The chip design process is inherently based on abstracting details of the design, and this is essential for complexity management. How can physics-based reliability models be used to analyze and optimize systems at higher levels? This talk will attempt to provide partial answers in this direction by discussing modeling methods for device-level failure mechanisms such as bias temperature instability, hot carrier injection, and gate oxide breakdown, where a great deal of advanced research has been carried out in the device community, but has not yet percolated far beyond.

3.23 Analysis of Field Test Effectiveness to LSI Reliability

Yasuo Sato (Kyushu Institute of Technology, JP)

License Creative Commons BY-NC-ND 3.0 Unported license
© Yasuo Sato

Potential of field errors caused by LSI degradation such as NBTI, HCI or so on is increasing. These errors essentially look different from the conventional errors such as permanent errors or soft errors. It means that the conventional dependability theories might not well reflect their impacts on safety systems. The author analyzes their impacts on safety systems and tries to find a proper index, which shows the effectiveness of the concept of proposed field test DART. DART (Dependable Architecture with Reliability Testing) repeatedly measures the maximum delay of LSI and monitors delay margin in field. Using this approach, delay degradation can be detected before it will cause an actual system error. The technology detail of DART is not discussed, but the relevance to the problem and what technologies should be developed by research people will be discussed.

3.24 How to Efficiently Analyze Aging Effects in Large Circuits - and Some Ideas How to Use the Results

Ulf Schlichtmann (TU München, DE)

License Creative Commons BY-NC-ND 3.0 Unported license
© Ulf Schlichtmann

Aging effects such as NBTI, PBTI, HCI are becoming more relevant as process technologies continue to scale. To date, commercial EDA tools only support the analysis of aging effects on transistor levels, thus severely limiting the size of designs that can be analyzed. We propose first a technique to efficiently analyze aging on gate level. We then introduce the concept of "potentially critical paths (PCPs)" which allows us to take the modeling of aging higher to the module level. We show how the concept of PCPs results in a further speedup of about 30x, without any loss in accuracy. Finally, we present some more ideas how the PCP concept can be used for further applications.

3.25 Towards Beneficial Hardware Acceleration of Functional Verification

Marcela Simkova (Brno University of Technology, CZ)

License Creative Commons BY-NC-ND 3.0 Unported license
© Marcela Simkova

Functional verification is a widespread technique to check whether a hardware design satisfies a given correctness specification. It is typically used in the pre-silicon phase of the design cycle to verify not only functional aspects but also reliability and safety properties. However, after the system is manufactured there are often found some previously uncovered errors. Moreover, further errors can be introduced by synthesis, mapping, place and route or fabrication processes. In order to eliminate as many remaining bugs as possible before a device is fabricated, verification is currently applied even in the post-silicon phase of the design cycle. Unfortunately, it is not possible to directly use the techniques from the pre-silicon phase (stimuli generation, assertion and coverage analysis, scoreboarding), and it is a challenging task to come up with techniques for post-silicon verification that would have strength comparable to the pre-silicon ones. In the presentation, I will talk about how to handle the gap between pre- and post-silicon verification using hardware acceleration with functional verification features. Furthermore, I will present HAVEN, an open framework for hardware acceleration of functional verification that provides means for seamless transition from pre- to post-silicon verification.

3.26 The Reliability Challenge from Random Process Variability Induced Timing Errors

Adit Singh (Auburn University, US)

License 🅭 🅮 ⊜ Creative Commons BY-NC-ND 3.0 Unported license
© Adit Singh

Current state-of-the-art timing test methods are not very effective in detecting delay faults in complex integrated circuits. Thus far, this has not been a major problem because manufacturing defects that cause subtle "delay only" failures are rare; they appear to be at least one to two orders of magnitude less frequent than defects that manifest as more easily detectable slow speed DC failures. Even if a significant fraction of delay defects remain undetected during production testing, the DPM impact on all but the lowest yielding ICs is generally quite modest. However, random transistor threshold voltage variation in aggressively scaled nanometer technologies is introducing a new source of timing variability. This variation is further amplified at the low operating voltages necessary for power minimization. Given the large number of transistors in a chip, hundreds of random "delay defects" can be statistically expected in every manufactured part in end-of-roadmap CMOS technologies. Moreover, because the increase in delay caused by a statistically slow transistor is potentially unbounded, each IC will need to be carefully tested for timing and reliably speed binned for use. This is a different and more formidable problem than the traditional speed binning of processors which is mostly aimed at handling systematic process variations –there are concerns whether delay test methodologies will be up to the task. The incorrect assignment of a higher speed to an IC because of improperly tested slow paths can result in operational failures in the field from timing errors. We discuss the significance of this emerging reliability challenge, and test and fault tolerance methods needed to address it.

3.27 Reliability Evaluation in Complex Systems: Some Cases of Study and Related Lessons

Matteo Sonza Reorda (Politecnico di Torino, IT)

License 🅭 🅮 ⊜ Creative Commons BY-NC-ND 3.0 Unported license
© Matteo Sonza Reorda

Assessing the reliability of complex systems is a challenging task. The talk will provide a couple of examples where this task has been performed, with details about the environments, results, and difficulties. Lessons will be drawn and open issues highlighted.

3.28 Wearout Modeling and Mitigation at Higher Levels of Abstraction

Mehdi B. Tahoori (KIT - Karlsruhe Institute of Technology, DE)

License 🅭 🅮 ⊜ Creative Commons BY-NC-ND 3.0 Unported license
© Mehdi B. Tahoori

As CMOS technology enters in nanoscale regimes, the reliability of VLSI chips is threatened by various issues such as increased process variation, radiation-induced soft errors, as well as transistor and interconnect aging. For cost-efficient resilient system design, reliability issues must be addressed at various design steps, together with other design objectives. In this talk, I will discuss some approaches to model and mitigate wearout, mostly due to transistor aging, at architecture level and early design stages. By considering reliability together with performance, cost, power objectives, it would be possible to balance them in a cost-effective way.

3.29 Validating Fault Tolerant Designs in SRAM-Based FPGAs: How to Keep the Route in an Ocean of Bits

Massimo Violante (Politecnico di Torino, IT)

License Creative Commons BY-NC-ND 3.0 Unported license
© Massimo Violante

SRAM-based FPGAs are more and more attractive for applications like avionic subsystems and satellite payloads. However, due to the lack of widely usable fault tolerant SRAM-based devices, it is up to the designer to implement suitable fault tolerant designs. The validation of such designs can be challenging, especially when considering single event upset in the device multi-million bits configuration memory. In this talk, a methodology will be illustrates to help designers to keep the proper route when validating fault tolerant circuits against the soft errors that may affect the ocean of bits in the configuration memory of SRAM-based devices.

3.30 Designing a Dependable Network-on-Chip Platform for Automotive Applications

Tomohiro Yoneda (NII - Tokyo, JP)

License Creative Commons BY-NC-ND 3.0 Unported license
© Tomohiro Yoneda

Current automotive electronic systems contain many *Electronic Control Units* (ECUs), and their functional safety is an important issue. We have been working to develop a dependable network-on-chip platform for implementing many ECUs on it. This talk will first introduce the designs for dependability in several different levels, such as circuit level, routing algorithm level, and processor core level, adopted in this platform. Then, several issues related to requirements and metrics specified in ISO26262 international standard on functional safety for road vehicles will be discussed, and a trial evaluation of our platform will be shown. Finally, a plan for the functional verification of the platform based on HIL (Hardware In the Loop) simulation will be introduced.

4 Panel Discussions

4.1 Beyond the Limitations of Approximate and Statistical Computing

Chair: Suddhakar M. Reddy
Panelists: J.P. Hayes, M. Orshansky, A. Raghunathan, S. Sapatnekar, A. Singh

Questions discussed during this panel where:

1. Why does approximate computing work?
2. Why does not work?
3. Where will it be applied and will there ever be appropriate devices?
4. Is there a design methodology to control the bound and to make guarantees?
5. Is it scalable?
6. What is the fine print in the trade-off approximation versus efficiency?
7. How is aging handled and predicted?

The main outcome of the panel was on the verification of approximate or probabilistic systems. The complexity increases tremendously if the whole system is considered with all the details. However, we cannot separate layers easily as this is typically done in traditional equivalence checking. Thus a neat methodology for the verification of approximate computing systems or probabilistic systems is required. This methodology must provide mechanisms for abstraction from one level to the next in order to control the complexity of the verification task.

4.2 What's Most Urgent in Industry?

Chair: Shawn Blanton
Panelists: R. Aitken, E. Arbel, C. Gebauer, B. Hoppe, Y. Sato

This panel was conducted in a round-robin fashion with one prime question targeted to each panelist: “What is the biggest issue you see?”. Sveral issues were raised in the panel.

First of all reliability is hard to sell as it does not manifest as an obvious feature to the customer. Customers do not want to spend extra space, i.e. money, to increase reliability in consumer electronics. This has to change in the future. Appropriate metrics may be the key point here.

Once additional cost is accepted, the result typically has to be exact and the data must be coherent on the functional level. Thus, hardware needs to be able to detect errors, i.e., avoiding *Potential Unexpected Loss of Data* (PULD). These features are required within a high reliable system. Then only correct data exchange to the environment needs to be ensured. An orthogonal design issue is mixed-mode operation of reliable computing applications and non-reliable computing applications that run in a single system. Some kind of “address space separation” will be required to guarantee reliability in this case.

On the verification side, verifying reliability still requires a lot of manual work, we need to have more powerful engines which can “reverse engineer” design intent, that is understand automatically how the protection in hardware works and be able to verify that it works correctly. And can we have a "killer-algorithm" which is able to verify all sorts of error detection and correction schemes in a generic way?

4.3 Fault Models, Metrics & Engines

Chair: Jacob A. Abraham,
Panelists: C. Braunstein, Y. Kinoshita, U. Schlichtmann, S. Blanton, M. Tahoori

Questions discussed during this panel where:

1. How do we generate good fault models, especially for determining the "faulty" behavior of an application (a.k.a. How do we get rid of faulty fault models?)
2. What metrics should we use, why, and how are they validated?
3. What engines would allow the calculation of validated reliability metrics for the entire system?

Some fault model free approaches have been proposed to make fault modeling more easy, but these are typically too conservative, i.e., they allow for situations that are virtually impossible in a real system. So mostly for new technologies fault models will still be determined by empirical studies. Engines that can classify a system or determine metrics will depend on the type of system. The co-existence of formal approaches and simulation techniques will continue.

Participants

- Jacob A. Abraham
Univ. of Texas at Austin, US
- Robert Aitken
ARM Inc. - San Jose, US
- Eli Arbel
IBM - Haifa, IL
- Bernd Becker
Universität Freiburg, DE
- Shawn Blanton
Carnegie Mellon University - Pittsburgh, US
- Cecile Braunstein
UPMC - Paris, FR
- Mehdi Dehbashi
Universität Bremen, DE
- Giuseppe Di Guglielmo
Università degli Studi di Verona, IT
- Rolf Drechsler
Universität Bremen, DE
- Görschwin Fey
Universität Bremen, DE
- Masahiro Fujita
University of Tokyo, JP
- Carsten Gebauer
Robert Bosch GmbH - Schwieberdingen, DE
- Jie Han
University of Alberta, CA
- John P. Hayes
University of Michigan, US
- Bodo Hoppe
IBM Deutschland - Böblingen, DE
- Ravishankar K. Iyer
University of Illinois - Urbana, US
- Seiji Kajihara
Kyushu Institute of Technology, JP
- Yoshiki Kinoshita
AIST - Hyogo, JP
- Wenchao Li
University of California - Berkeley, US
- Igor L. Markov
University of Michigan, US
- Yusuke Matsunaga
Kyushu University, JP
- Subhasish Mitra
Stanford University, US
- Michael Orshansky
Univ. of Texas at Austin, US
- Laurence Pierre
TIMA - Grenoble, FR
- Ilia Polian
Universität Passau, DE
- Anand Raghunathan
Purdue University, US
- Sudhakar M. Reddy
University of Iowa - Iowa City, US
- Kaushik Roy
Purdue University, US
- Sachin Sapatnekar
University of Minnesota, US
- Yasuo Sato
Kyushu Institute of Technology, JP
- Matthias Sauer
Universität Freiburg, DE
- Ulf Schlichtmann
TU München, DE
- Marcela Simková
Brno University of Technology, CZ
- Adit Singh
Auburn University, US
- Matteo Sonza Reorda
Politecnico di Torino, IT
- Mehdi B. Tahoori
KIT - Karlsruhe Institute of Technology, DE
- Massimo Violante
Politecnico di Torino, IT
- Tomohiro Yoneda
NII - Tokyo, JP