A Strategy for Reliability Assessment of Future Nano-Circuits

Sanja Lazarova-Molnar, Valeriu Beiu, and Walid Ibrahim
College of Information Technology
United Arab Emirates University
PO Box 17555, Al Ain
UNITED ARAB EMIRATES
http://www.cit2.uaeu.ac.ae

Abstract: - This paper summarizes a strategy for development of an EDA (Electronic Design Automation) tool which is aimed to support the design of future nano-circuits. The problem with the existing EDA tools is that they do not explicitly consider reliability as a design criterion. Most of the tools that do consider reliability are not intended for the nanoelectronic industry and are very limited in the types of failure models they can assess. Moreover, current indications show that moving towards nano-scale will significantly increase the failure rates. It follows that an improved EDA tool which would efficiently assess reliability (besides speed, power, area, etc.) is becoming a necessity. In this paper we detail a strategy and its methods that could ultimately lead to an EDA tool for realistic reliability evaluation of nano-circuits.

Key-Words: - Nanoelectronics, Reliability, EDA, Proxels, Simulation, Fault model.

1 Introduction

The expectations are that the future nano-circuits will exhibit higher frequency of failures. The higher density of transistors on chip is one of the reasons for this behavior. Other factors that impact the reliability are the geometric variations and manufacturing defects. This implies that reliability should be included as a fourth optimization parameter in the future Electronic Design Automation (EDA) tools. The current optimization parameters are: area, speed and power.

The reliability evaluation tools currently available can be classified as either special or general purpose tools. The special-purpose tools are EDA tools designed specifically to evaluate the reliability of electronic circuits. The most popular tools in this class are NANOPRISM [1] and RAMP [2]. NANOPRISM is a probabilistic model checking based tool, which was developed at Virginia Polytechnic University. It uses model checking techniques for calculating probabilities of transient failures in the devices and interconnections of nanoarchitectures. RAMP has two implementations, 1.0 and 2.0, which are different in both efficiency and the assumptions they impose on the analyzed models. RAMP 1.0 is simpler and can be applied both to real hardware and used in simulators. RAMP 2.0 allows for more complex models to be analyzed and uses the Monte-Carlo method to run experiments. However, it cannot be applied to real hardware.

The second category of tools encompasses general reliability tools, which are very many. The drawback of most of them is that they are not specifically aimed at analyzing circuits and each will have to be manually adapted to respond to the modeling needs of nanoelectronics. In the following we summarize a few of them.

The Hybrid Automated Reliability Predictor (HARP) tool was pioneered in 1981 at Duke and Clemson University. HARP uses a fault-tree analysis technique for describing the failure behavior of complex technical systems. Fault tree diagrams are logical block diagrams that display the state of a system in terms of its components. The basic elements of the fault tree are usually failures of different components of one system. The combination of these failures determines the failure of the system as a whole. Further development have led to Symbolic Hierarchical Automated Reliability and Performance Evaluator (SHARPE) [3] (Duke University) and Monte Carlo Integrated HARP (MCI-HARP) [4] (developed at Northeastern University).

In the early 90s a few other tools providing numerical analyses have been developed: TimeNET at the Technical University of Berlin (pdv.cs.tu-berlin.de/~timenet/), UltraSUN (and later Möbius) at the University of Illinois at Urbana-Champaign, and SMART at the University of California at Riverside. These were followed in the mid-90s by Dynamic Innovative Fault Tree (DIFTree) [5], and Galileo [6], both from the University of Virginia. Galileo

extended the earlier work on HARP, MCI-HARP and DIFTree using a combination of binary decision diagrams (BDD) and Markov methods, and is currently being commercialized by Exelix.

In 1999 a team from the University of Birmingham introduced the Probabilistic Symbolic Model Checker (PRISM) [7]. PRISM relies on a probabilistic model checking for determining if a given probabilistic system satisfies given probabilistic specifications. It applies algorithmic techniques to analyze the state space and calculate performance measures associated to the probabilistic model. PRISM supports the analysis of DTMCs, CTMCs, and Markov decision processes (MDPs).

The probabilistic transfer matrices (PTMs) framework was first presented in [8], but the underlying concept can be traced back to [9]. The PTMs can be used to evaluate the circuit overall reliability by combining the PTMs of elementary gates or sub-circuits [10]. It performs simultaneous computation over all possible input combinations, and calculates the exact probabilities of errors. Another advantage (beside accuracy) is that it is trivial to have different probabilities of failures for the different gates (see [11]). PTM however has a major memory bottleneck: for a circuit with n inputs and m outputs, the straightforward PTM representation requires O(2n+m) memory space. This limits the size of the circuits that can be simulated to about 16 input/output signals.

Recent work has also been done in modeling signal dependencies using Bayesian Networks (BNs) [12]. The relation between circuit signals and Markov random fields was presented in the context of probabilistic computations. The conditional probability of output(s) given input signals determines how errors are propagated through a circuit. Using this theoretical model, it is possible to predict the probability of output error given the gate errors.

The main problem with most of the existing approaches is that they assume exponential distribution of devices' and gates' failures. This means that the probabilities of failure are independent of the length of time the gates and devices have been in use. Conversely, it has been shown that the exponential behavior of failures is incorrect and produces significant error that cannot be ignored [13].

Our goal is to develop an EDA tool that would overcome some of the problems of the existing tools, including the most significant one of over-simplifying the fault models. In this paper we present the strategy and the methods which will be used for that purpose.

2 What is Our Strategy?

The EDA tool that we plan to develop would enable a more accurate reliability evaluation. It will allow designers to evaluate and compare the reliability of different nano-architectures and select the best architecture that meets the target area, speed, power as well as reliability requirements.

The following four tasks can be identified as main components of our strategy:

1) Fault models acquirement.
2) Design of reliability evaluation algorithms.
3) Development of an EDA tool for reliability evaluation.
4) Choice of validation strategy and performance evaluation.

Further we describe each of the tasks.

2.1 Fault Models Acquirement

Accurate fault modeling at both device and gate levels is essential for successful reliability estimation. There are currently almost no models that can be used to precisely estimate the manufacturing defects or transient error rates in future nano-devices [14], [15]. It was even mentioned that existing fault models might be reevaluated or completely discarded [16].

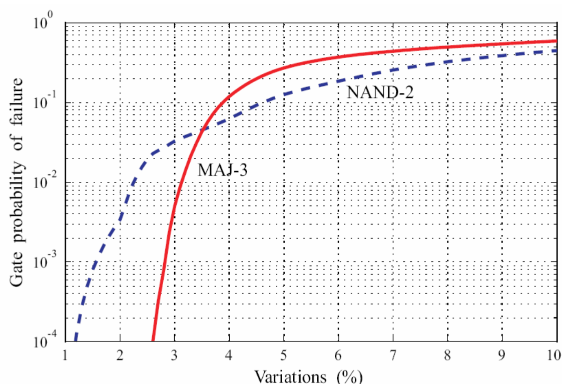

Fig. 1. Dependence of the gate probability of failure on geometric variations for single electron technology

Most of the current literature on VLSI fault modeling assumes that devices or gates have a constant probability of failure. Thus, we need to start by generating a collection of new fault models for nano-circuits that are as close as possible to their real behavior (see Fig. 1). This will represent the first phase and the starting point of our project.

The errors that can appear at nano-scale level are classified in two categories, i.e. soft and hard errors. Soft errors occur mostly due to noise or external radiation. They are also known as transient errors because the circuit usually recovers from them. Hard errors can be classified as either extrinsic or intrinsic. Extrinsic errors are basically manufacturing errors (also known as defects), which mostly appear from the very beginning; whereas intrinsic errors appear due to wear-out, i.e. aging of the components. Our intent is to observe both types of errors simultaneously, and therefore we will need to collect data for modeling both of them.

The fault modeling will be accomplished by means of time-consuming Monte-Carlo simulations. We will use random numbers to reproduce the different types of variations/noise/etc. at device level, and then perform simulations to estimate the device probability of failure (corresponding to such variations). The results obtained using Monte-Carlo simulations will be fitted using polynomial interpolation in order to prepare them for the proxel-based simulation [17], [18] (see Fig. 2).

2.2 Design of Reliability Algorithms

Once the device fault modeling phase is completed, the next step will be to evaluate the gates' reliability using the proxel-based method. The proxel-based method can address the errors due to aging adequately and will not oversimplify them to a constant probability of failure.

The proxel-based method was introduced in 2002 [17] as an alternative to Monte-Carlo for simulating discrete stochastic models. Borrowing from pixel, proxel is the abbreviation of "probability element." It describes every probabilistic configuration of the model in a minimal and complete way. Each proxel carries enough information for generating its successor proxels, i.e., for determining probabilistically how the model will behave [18]. This transforms a non-Markovian model into a Markovian one. This approach analyzes models in a deterministic manner, avoiding the typical problems of Monte-Carlo simulation (e.g., finding good-quality pseudo-random-number generators) and partial differential equations (PDEs, difficult to set-up and solve). The underlying stochastic process is a discrete-time Markov chain (DTMC), which is constructed on-the-fly by inspecting all possible behaviors of the model.

The proxel-based method combines the benefits of both Monte-Carlo and PDEs [18]. It has a high modeling capacity, which is equivalent to the Monte-Carlo one, and it does not rely on generating random numbers. Additionally, it does not impose a constant failure rate assumption on the models, that pertains to some numerical approaches.

We plan to exploit the first-error accuracy of the proxel-based method for obtaining accurate solutions within relatively short computation times [18]. This will translate into obtaining fast and good reliability estimates.

As shown in [19], the proxel-based method can be extended to include various factors for reliability estimation, allowing for quite complex models to be analyzed. These will more closely reflect reality. The possibility to include other parameters (e.g., temperature) is a feature allowed by the highly flexible definition of a proxel. This also implies that the method could be extended to include both soft and hard errors, and make them both an integral part of our novel and enabling reliability estimation EDA process. This will allow for a comprehensive observation of the reliability of nano-circuits.

As shown in [19], the proxel-based method can be extended to include various factors for reliability estimation, allowing for quite complex models to be analyzed. These will more closely reflect reality. The possibility to include other parameters (e.g., temperature) is a feature allowed by the highly flexible definition of a proxel. This also implies that the method could be extended to include both soft and hard errors, and make them both an integral part of our novel and enabling reliability estimation EDA process. This will allow for a comprehensive observation of the reliability of nano-circuits.

The combination of both types of errors for reliability evaluation is something that, to the best of our knowledge, has never been done before. Our positive expectations regarding the success of our approach are based on the fact that the proxel-based method is extremely flexible. As long as the changes over time can explicitly be described in terms of the state variables of the system, the proxel-based method can simulate them.

In addition, we have successfully applied the proxel-based to performability analysis [20] of small-scale models [19], as well as to a warranty analysis problem for the automotive industry [21]. For the second case the speed-up factor achieved was a whopping 1500x. This has reduced computation times from about one day (when using Monte-Carlo) to about one minute (when using proxels).

2.3 EDA Tool Development

The EDA tool we plan to develop will be designed to provide users with a friendly access to the enabling fault models and novel reliability algorithm. The EDA tool task is highlighted by the

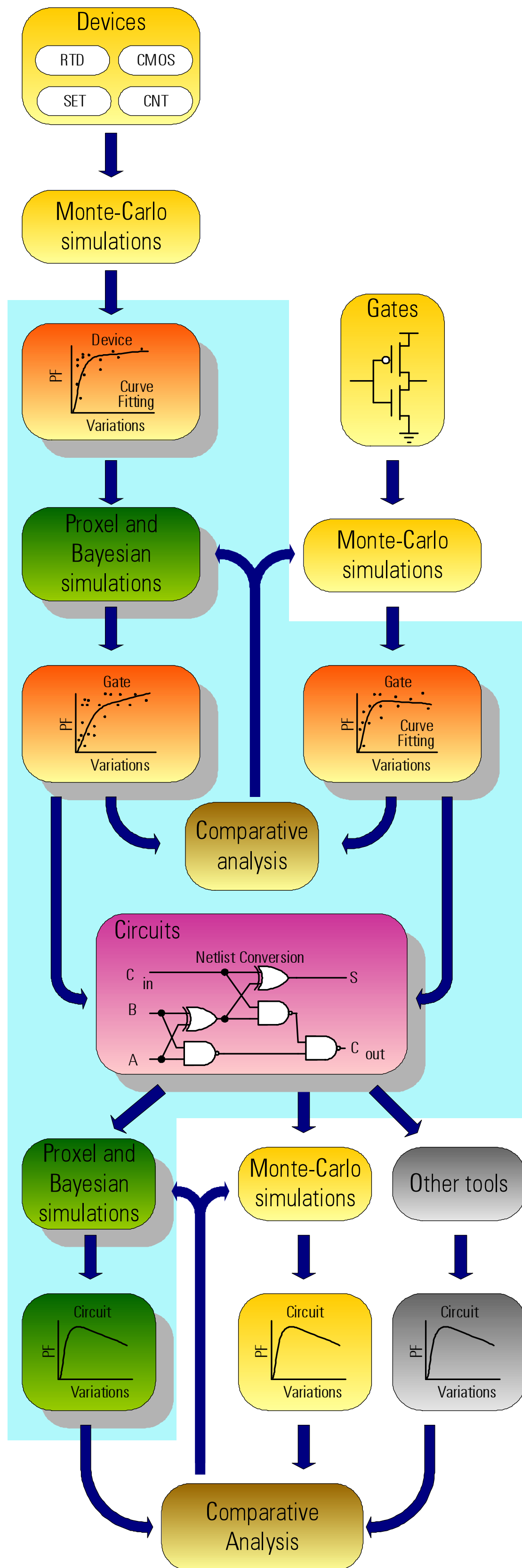

Fig. 2. Flowchart showing tasks envisioned and their results

blue background in Fig. 2, and encompasses: curve fitting, proxel simulations, netlist conversion, results comparison, and reporting tasks. The first curve-fitting task will convert the results obtained from Monte-Carlo simulations into simpler polynomial functions. This task is essential as providing the proxel algorithm with the "proper inputs" (i.e., the variable probabilities of failure of the devices resulting from Monte-Carlo simulations) for evaluating the gate's probability of failure. The second curve-fitting task will convert the gates' probability of failure results obtained from Monte-Carlo simulations into simpler polynomial functions. The outputs from this task will be used to check the accuracy of the proxel algorithm applied at the gate level (i.e., benchmarking them against the Monte-Carlo simulations). This will allow us to fine-tune the proxel algorithm before going to the next level: the circuit level.

In order to evaluate the reliability of a given circuit, a complete description of the circuit is required. Such descriptions normally include the number of gates, the type of gates, fan-in and fan-out for each gate, the location of each gate (relative to other gates), as well as the gates' interconnections. This information will have to be converted to the data structure required by the reliability evaluation algorithm (e.g., proxel-based). Converting the circuit description to the suitable data format manually might be an easy task for small circuits. However, this is a daunting and error-prone task in case of slightly larger circuits. That is why, the EDA tool has to provide users with an automatic and simple way to bridge the data formatting gap.

Instead of supplying the circuit description in the format required by the proxel-based method, an automated conversion tool will be developed. This will accept as input a circuit description in the standard netlist (a list of logic gates and their interconnections which make up a circuit) format. The netlist conversion task will generate the (proper) data structure required by the reliability evaluation algorithm. This will also allow the future expansion of our EDA tool by seamless integration.

The EDA tool will provide users with a user friendly GUI to evaluate and compare the reliability results of different circuits. The GUI can also be used to find the operating conditions for a circuit in order to satisfy predefined reliability constraints. For instance, the user could find the maximum allowed manufacture variations, or the maximum allowed temperature fluctuations, such that the reliability of the designed circuit is above 99.99%. Obviously, conversion to failure in time (FIT) will be available

automatically.

2.4 Validation Strategy and Performance Evaluation

One of the most difficult tasks facing software designers is the evaluation and comparison of different tools and algorithms. Software designers should be able to measure the efficiency (speed and memory requirements) as well as accuracy of different algorithms, and compare them (both for simple and for complex input data) in order to understand both their behavior and their progress over time. Over the last few decades, there have been many attempts to create and use neutral benchmarks for tool evaluation and comparison. Typically, a benchmark set consists of a collection of problems in a common format, which attempts to represent a wide range of inputs for evaluating algorithms and tools. Obviously, benchmarks are specific for a certain domain. Still, if everyone uses the same test cases to evaluate similar tools, it should be straightforward to compare results.

In the VLSI community there are several benchmark sets which are widely used. The reliability EDA tool will be evaluated using the International Symposium on Circuits and Systems (ISCAS) benchmarks. These benchmarks consist of collections of circuits contributed by a number of individuals and organizations over a period of years. The origins of the first test set go back to a special session of ISCAS'85, which brought together nine research teams presenting experimental results of combinational test generation algorithms on 10 circuits that were distributed to each team in advance.

The performance of our proposed algorithm and fault models will be compared against the performance of commercially available reliability evaluation tools (e.g., Relex Reliability Studio and Reliass) and other algorithms (Probabilistic Transfer Matrices, Probabilistic Gate Models, and Bayesian Networks). The comparison criteria will include: accuracy, speed, and memory requirements. Since most of these tools cannot handle the case where the gate's probability of failure is variable, for a fair comparison several predefined constant probabilities of failure will be used for the performance evaluation purposes.

3 Summary and Outlook

It is expected that nano-devices will be highly unreliable [22]. Therefore, EDA tools for reliability evaluations will be essential in order to help VLSI designers develop applications that meet the size and power specifications, while still being reliabile enough.

This paper describes a high level approach to encounter one of the biggest challenges of future nano-circuit design, namely reliability. Our strategy emphasizes the importance of reliability and argues for the development of an enabling EDA tool. The EDA tool would respond to the new requirements for realistic reliability evaluations by employing a data collection that will not over-simplify the models. This implies that, the simulation methods need also to be flexible enough, and with controllable accuracy. This is reflected in our choice of simulation method, i.e. the proxel-based method.

Our intention is to develop several accurate fault models (at both the device and the gate level). These models will be among the first ones to consider variable probabilities of failure (w.r.t. process fluctuations, temperature variations, etc.). We expect that the developed EDA tool will be more precise, faster, and will require less memory than EDA reliability tools currently available.

References:

[1] D. Bhaduri. and S. Shukla, "Nanoprism: A tool for evaluating granularity vs. reliability trade-offs in nano-architectures," Proc. GLSVLSI'04, Boston, USA, ACM, April 2004.

[2] J. Srinivasan, "Lifetime reliability aware microprocessors," Ph.D. dissertation, Univ. of Illinois at Urbana-Champaign, May 2006. Available at: http://rsim.cs.uiuc.edu/Pubs/srinivsn-phd-thesis.pdf

[3] R.A. Sahner, and K.S. Trivedi, "Reliability modeling using SHARPE," IEEE Trans. Reliability, vol. 13, Jun. 1987, pp. 186–193.

[4] M.A Boyd, and S.J. Bavuso, "Simulation modeling for long duration spacecraft control systems," Proc. Annual Reliability & Maintainability Symp., Atlanta, GA, USA, Jan. 1993, pp. 106–113.

[5] J.B. Dugan, B. Venkataraman, and R. Gulati, "DIFTree: A software package for the analysis of dynamic fault tree models," Proc. Annual Rel. & Maintainability Symp., Philadelphia, PA, USA, Jan. 1997, pp. 64–70.

[6] D. Coppit, and K.J. Sullivan, "Galileo: A tool built for mass-market applications," Proc. Intl. Conf. Software Eng., Limerick, Ireland, Jun. 2000, pp. 273–282.

[7] M. Kwiatkowska, G. Norman, D. Parker, and R. Segala, "Symbolic model checking of concurrent probabilistic systems using MTBDDs and

simplex," Tech. Rep. CSR-99-01, School of Comp. Sci., Univ. of Birmingham, Birmingham, UK, Jan. 22, 1999. Available at: http://www.cs.bham.ac.uk/~dxp/papers/CSR-99-01.pdf

[8] K.N. Patel, I.L. Markov, and J.P. Hayes, "Evaluating circuit reliability under probabilistic gate-level fault models," Proc. Intl. Workshop Logic Synthesis IWLS'03, Laguna Beach, CA, USA, May 2003, pp. 59–64.

[9] V.L. Levin, "Probability analysis of combination systems and their reliability," Eng. Cyber., vol. 6, Nov-Dec. 1964, pp. 78–84.

[10] S. Krishnaswamy, G.F. Viamontes, I.L. Markov, and J.P. Hayes, "Accurate reliability evaluation and enhancements via probabilistic transfer matrices," Proc. Design Autom. & Test Europe DATE'05, Munich, Germany, Mar. 2005, pp. 282–287.

[11] W. Ibrahim, V. Beiu, and Y. A. Alkhawwar, "On the reliability of four full adder cells," Proc. Intl. Design & Test Workshop IDT'06, Dubai, UAE, Nov. 2006, in press.

[12] T. Rejimon, and S. Bhanja, "An accurate probabilistic model for error detection," Proc. Intl. Conf. VLSI Design VLSID'05, Kolkata, India, Jan. 2005, pp. 717–722.

[13] V. Beiu, W. Ibrahim, Y. A. Alkhawwar, and M. H. Sulieman, "Gate Failures Effectively Shape Multiplexing," Proc. IEEE Intl. Symp. on Defect & Fault Tolerance in VLSI Sys. DFT'06, Washington, USA, Oct. 2006, pp. 29–40.

[14] M. Forshaw, R. Stadler, D. Crawley, and K. Nicolić, "A short review of nanoelectronic architectures," Nanotechnology, vol. 15, Feb. 2004, pp. S220–S223.

[15] M. Hartmann, and P. C. Haddow, "Evolution of fault-tolerant and noise-robust digital designs," IEE Proc. Comp. & Digital Tech., vol. 151, Jul. 2004, pp. 287–294.

[16] J.A.B. Fortes, "Future challenges in VLSI system design," Proc. Intl. Symp. VLSI ISVLSI'03, Tampa, USA, Feb. 2003, pp. 5–7.

[17] G. Horton, "A new paradigm for the numerical simulation of stochastic Petri nets with general firing times," Proc. European Simulation Symp. ESS'02, Dresden, Germany, Verlag, Oct. 2002. Available at http://www.scs-europe.net/conf/ess2002/meth-20.pdf

[18] S. Lazarova-Molnar, "The proxel-based method: Formalisation, analysis and applications," Ph.D. dissertation, Otto-von-Guericke Univ. of Magdeburg, Germany, Nov. 2005. Available at: http://diglib.uni-magdeburg.de/Dissertationen/2005/sanlazarova.pdf

[19] S. Lazarova-Molnar, and G. Horton, "A framework for performability modelling using proxels," Proc. ICMSAO'05, Sharjah, UAE, Feb. 2005,.

[20] B. Haverkort, R. Marie, G. Rubino, and K. Trivedi, Performability Modelling: Techniques and Tools, John Wiley & Sons, 2001.

[21] S. Lazarova-Molnar, and G. Horton, "Proxel-based simulation of a warranty model," Proc. European Sim. Multiconf. ESM'04, Magdeburg, Jun. 2004, pp. 221–224.

[22] International Technology Roadmap for Semiconductors, ITRS 2005. Available at: http://public.itrs.net/