A Bayesian-Based EDA Tool for Nano-circuits Reliability Calculations

Walid Ibrahim* and Valeriu Beiu

College of Information Technology, UAE University
P.O. Box 17551, Al Ain, UAE
{walidibr,vbeiu}@uaeu.ac.ae

Abstract. As the sizes of (nano-)devices are aggressively scaled deep into the nanometer range, the design and manufacturing of future (nano-)circuits will become extremely complex and inevitably will introduce more defects while their functioning will be adversely affected by transient faults. Therefore, accurately calculating the reliability of future designs will become a very important aspect for (nano-)circuit designers as they investigate several design alternatives to optimize the trade-offs between the conflicting metrics of area-power-energy-delay versus reliability. This paper introduces a novel generic technique for the accurate calculation of the reliability of future nano-circuits. Our aim is to provide both educational and research institutions (as well as the semiconductor industry at a later stage) with an accurate and easy to use tool for closely comparing the reliability of different design alternatives, and for being able to easily select the design that best fits a set of given (design) constraints. Moreover, the reliability model generated by the tool should empower designers with the unique opportunity of understanding the influence individual gates play on the design's overall reliability, and identifying those (few) gates which impact the design's reliability most significantly.

Keywords: Reliability, Bayesian networks, EDA tools, nano-circuits.

1 Introduction

Nanotechnology is expected to be one of the fundamental technologies of 21st century. New nanomaterials and nanodevices will have a major impact in all areas of the global economy. It is expected to be applied across a wide range of industries and is inherently interdisciplinary, combining fields such as physics, chemistry, biology, computer science, and manufacturing. Some of the industries which are/will be impacted by nanotechnology include pharmaceuticals, medical, biotechnology, aerospace/aviation, defense, automotive, computers, semiconductors, information technology, communications, and energy. Due to its wide scope and impact, nanotechnology has the potential to create an industrial revolution that should have a huge influence on society and our everyday life, expectedly comparable to the information technology developments of the 20th century.

* Corresponding author.

A. Schmid et al. (Eds.): Nano-Net 2009, LNICST 20, pp. 276–284, 2009.

Analysts estimate that the market for nanotechnology products will increase to US$2.6 trillion by 2014 [1], and that ten million new jobs will be created in areas of manufacturing related to nanotechnology by then. Therefore, worldwide, governments and private companies have invested around US$6 billion in nanotechnology research and development in 2004. The figure has been raised to US$9.6 billion in 2005 and is expected to raise dramatically in the coming years, according to Angela Hullmann of DG Research, European Commission. Hundreds of nanotechnology-based products are already on sale, and many more are in the pipeline. About 65 countries, including the EU member states, Japan and the United States, as well as developing countries and emerging economies, are currently funding nanotechnology research heavily.

In the semiconductor sector, the nano-era has already started in 2002 with the introduction of the 90nm process by IBM. This was followed in 2006 by the 65nm in a variety of Intel microprocessors. In 2007 Matsushita and Intel started mass production of 45nm chips. Scaling the semiconductor technology deep into the nano-regime will lead to new classes of applications that include wireless sensor networks, wearable computers, implantable devices, etc. The application space is quickly becoming much larger, encompassing: ticketing, check-out counters, warehouse inventory tracking/management, shipping verification, location sensing, patient monitoring, machine-mounted sensing, building climate control, and security. The implantable systems in particular hold great promise for health applications: pace makers, defibrillators and hearing aids being already in use, while retina, ear, and neural implants are starting to be offered. The emergence of these real-life applications depends to a great extent on the ability to fabricate/manufacture *small, ultra low-power, highly reliable (electronic) circuits*.

The development of ever-smaller devices brings promise for further improvement in the performance of future integrated circuits (ICs)—reduce their size and power consumption—yet also leads to several new technical challenges, including *the need for architectures that reduce the uncertainty inherent to computations at very small scales* [1]–[4]. In particular, as feature sizes are aggressively scaled, the processing of ICs becomes more complex and inevitably introduces more defects. The devices' small sizes, and consequently the tiny amounts of energy required and allowing in their switching, makes them susceptible to transient failures [5], [6]. Architectures built from emerging nanodevices, such as nano-scale CMOS, SET (Single Electron Technology/Transistor), carbon nano tubes (CNT), silicon nanowires, molecular devices, spin transistor, etc., will be even more susceptible to parameter variations, fabrication defects, and transient failures induced by environmental/external causes [3], [4].

Therefore, the design community has been urging CAD researchers to pay more attention to reliability issues. This was the message from the ICCAD'06 conference, which gathered together both communities (design and CAD) in San Jose, in November 2006. The commercially available tools that feature design-for-reliability do not consider the delay and power requirements simultaneously (according to design engineers present at the conference). Consequently, there is an pressing need for adequate reliability assessment tools that would satisfy the trade-off between the contradicting delay, power, and reliability constraints. In the rest of the paper we will review of the state-of-the-art of reliability tools in Section 2. The proposed methodology for a reliability enabled EDA tool is presented in Section 3, followed by conclusions and closing remarks in Section 4.

2 Reliability/Yield Estimations Tools

Calculating the reliability of a large (*i.e.*, complex) logic circuit can be done analytically (mathematical evaluations/equations, **EQ**), and/or based on simulation methods. The methods used for simulating stochastic systems can be divided further into experimental and numerical methods. In the case of experimental methods, the analysis is performed implicitly by observing the results obtained from many experiment runs.

The most popular experimental method is *Monte-Carlo simulation* (**MC**), which reproduces the behavior of the system. For doing this, Monte-Carlo relies on random number generators that sample the random activities of the system being analyzed. Once the model is built, the computer performs as many sample runs from the model as necessary to draw meaningful conclusions about the model's behavior. It follows that the Monte-Carlo based analysis is conducted indirectly, based on the observation of many sample runs. The biggest advantages of MC simulation are its intuitiveness, and its ability of simulating models for which deterministic solutions are intractable. *The MC method is by far the most widespread one in the semiconductor community* [7]. It will have to be used in the future for intimately analyzing the behavior of (novel) devices and gates as well as small sub-circuits [8]. Being (very) time consuming its use appears to be limited, but the reliability results obtained could be stored as parameters of future libraries (of devices and gates).

Numerical methods (algorithms, **ALG**) are designed for analyzing stochastic models without incorporating the random behavior. The simulation results that they deliver are the same for the same model parameters. These methods work by describing the flow of probabilities within the system—usually using differential equations and numerical methods for solving them. Markov chains can be used for describing and analyzing models that contain exclusively exponentially distributed state changes. Therefore, *one problem with numerical approaches is that they start from the assumption that the failure rate is constant* [9]. This has been shown to be incorrect and introduces significant errors that cannot be ignored [10]–[15]. Depending on the character of the time domain, there are discrete-time Markov chains (DTMCs) and continuous-time Markov chains (CTMCs). In the following we shall provide a brief chronological review of most of the numerical tools and techniques which have already been developed for reliability estimation/evaluation.

The *Hybrid Automated Reliability Predictor* (HARP) tool was pioneered in 1981 [16]. 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 developments have led to Symbolic Hierarchical Automated Reliability and Performance Evaluator (SHARPE) [17], [18] and Monte Carlo Integrated HARP (MCI-HARP) [19].

In the early 90s a few other tools providing numerical analyses have been developed: UltraSAN [20], Möbius [21], and SMART [22]. These were followed in the mid-90s by Dynamic Innovative Fault Tree (DIFTree) [23], and Galileo [24]. 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.

1999 saw the introduction of the Probabilistic Symbolic Model Checker (PRISM) [25]. PRISM relies on a probabilistic model checking for determining if a given probabilistic system satisfies given probabilistic specifications. The circuit is described as a state transition system with probabilities attached to each of the transitions. It applies numerical 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.

The *proxel* based method was introduced in 2002 [26] as an alternative to simulating discrete stochastic models. Borrowing from pixel, proxel is the abbreviation for "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 [27]. This transforms a non-Markovian model into a Markovian one. The approach analyzes models in a deterministic manner, avoiding the typical Monte Carlo problems (*e.g.*, finding good-quality pseudo-random-number generators) and partial differential equations (difficult to set-up and solve). The underlying stochastic process is a DTMC, which is constructed on-the-fly by inspecting all the possible behaviors of the model.

The probabilistic transfer matrices (PTMs) framework was first presented in [28], but the underlying concept can be traced back to [29]. The PTMs can be used to evaluate the circuit overall reliability by combining the PTMs of elementary gates or sub-circuits [30]. It performs simultaneous computation over all possible input combinations, and calculates the exact probabilities of errors. Another advantage (beside absolute accuracy) is that it is trivial to have different probabilities of failures for the different gates (see [12]). PTM however has a major memory bottleneck: for a circuit with n inputs and m outputs, the straightforward PTM representation requires $O(2^{n+m})$ memory space. This limits the size of the circuits that can be simulated to about 16 input/output signals. The use of the algebraic decision diagrams compression method can reduce the memory requirements, and circuits with about 40 input/output signals have been evaluated [30]. Our research group (Nano-ART) has started using PTM since 2005. We were able to use PTM to exactly calculate the reliability of several small circuits [11], [12]. We have developed an automated tool to extract the circuit information from Verilog netlist files and generate the Matlab files corresponding to the circuit's PTM [32]–[34].

Recent work has also been done in modeling signal dependencies using Bayesian Networks (BNs) [35]–[37]. 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.

It follows that any approach for estimating reliability has to be based on one or a combination of some of these *three different alternatives*: **EQ**, **ALG**, and **MC**. Each of these three methods can be applied at different levels, out of which the following *four levels* can easily be identified: (1) device, (2) gate (tens of devices), (3) circuit/core (thousands of gates), and (4) network (-on-chip), or many-/multi-cores. The three methods and the four levels lead to 39 different possible combinations

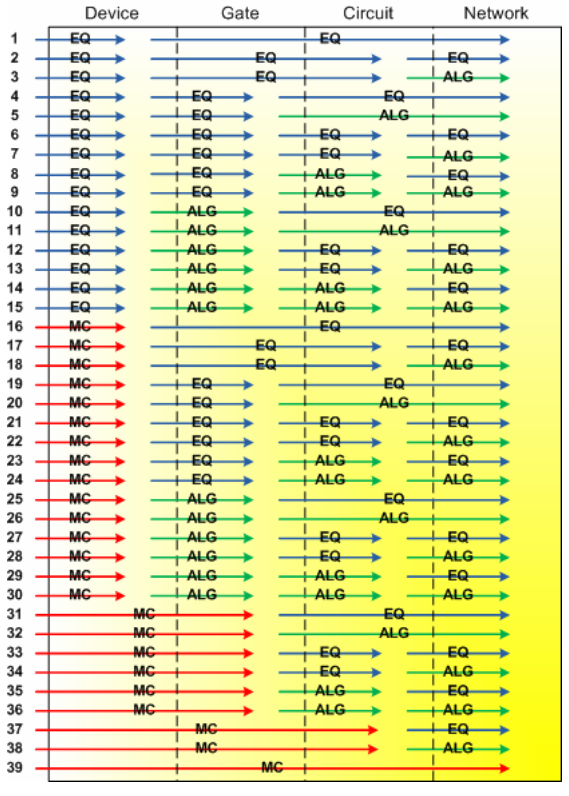

Fig. 1. Possible alternatives for reliability/yield estimations

(see Fig. 1). Unfortunately, very few of these 39 alternatives have been (or are being) used, with more than half of them never explored. Obviously, using **EQ** alone (alternative #1) is certainly the fastest approach (but not necessarily the most accurate one), while using **MC** up to the network level (alternatives #39) is certainly the most time consuming solution, but one which could lead to quite accurate results.

3 Reliability Enabled EDA Tool

In addition to their drawbacks, the above tools are mostly generic tools which were not designed or optimized for seamless integration in the mainstream of the semiconductor EDA tools. This section illustrates a generic architecture of an EDA tool for accurate calculation of future nano-circuits reliability (see Fig 2). The tool is divided into three main modules, which allow for reliability calculation at the device, the gate, and the circuit level.

3.1 Device Module

In recent publications [10]–[15], we have shown that accurate calculation of the reliability of future nano-circuits has to start from the device (transistor) level. Failing to

calculate/estimate the reliability starting from the device level, or simply assuming that all the gates have the same reliability (as customary done in the reliability literature), is unacceptable as leading to results which can be off by a few orders of magnitude. The proposed tool uses both Monte Carlo (MC) and numerical methods to capture the effects of scaled CMOS devices (deep into the nano-meter regime) on their reliability margins. It should also capture the effect of the expected (severe) V_{TH} variations on the devices' reliability [8], [38]. The aim of this module it to generate accurate device reliability models that can be used to quickly and precisely estimate the reliability of a given device for different materials and feature sizes.

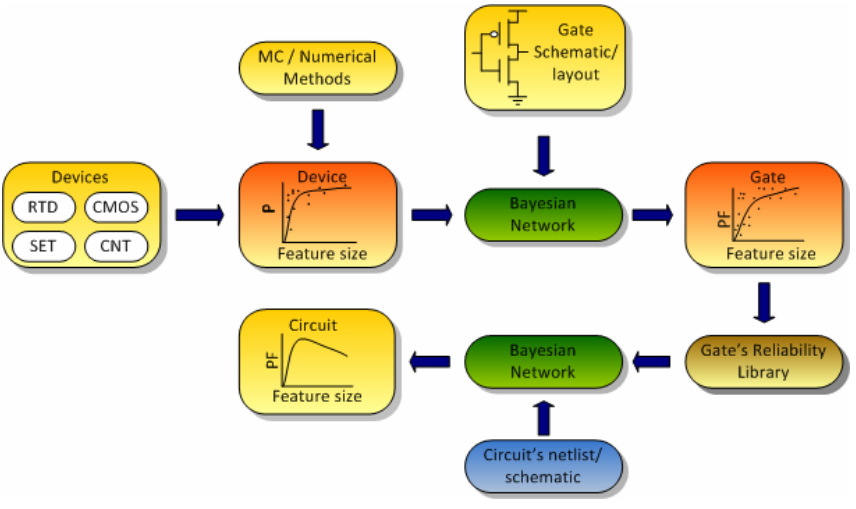

Fig. 2. Reliability enabled EDA tool architecture

3.2 Gate Module

The gate module uses Bayesian Networks (BN) to automatically construct reliability models for any gate starting from its elementary devices. The input to this module is the gate's schematic and the reliability models of the individual devices. The gate reliability models provide the circuit designers with the ability to thoroughly investigate the reliability of individual gates. They are particularly useful as they allow users to study the effect of increasing the reliability of individual devices (*e.g.*, different sizing of CMOS transistors or using radiation hardened designs) on the gate's reliability. The module compiles the generated reliability models into a reliability-enabled library of gates. The library contains multiple implementations for each gate together with their associated probabilities of failure.

3.3 Circuit/Core Module

To calculate the overall circuit's reliability, a tool needs a comprehensive description of the circuit's layout. This description should include a list of the circuit's gates as well as their types and interconnections. Preparing this information is a complex and

error prone process (especially for future Tera-scale circuits). Therefore, a reliability-enabled EDA tool should be able to automatically acquire this information for any circuit. The proposed tool parses the netlist file generated by most of the current EDA tools, to acquire the required information.

The proposed tool uses the extracted circuit/core layout information along with the reliability of the individual gates (from the reliability-enabled library of gates) to construct the circuit's BN reliability model. The constructed reliability model is then used by the tool to accurately calculate the circuit's overall reliability.

The tool should also consider the effect of the input vectors on the circuit's overall reliability. In addition to calculating the average overall reliability, the tool is able to identify the input vectors corresponding to the upper/lower reliability bounds [39].

Finally, the tool uses the BN reliability model to identify the set of gates that has the major impact on the circuit's reliability. Fine tuning the reliability of this set of gates (*e.g.*, by selecting different implementation from the reliability-enabled library of gates, or selecting different degrees, levels, or redundancy schemes) will/could significantly improve the trade-off between the conflicting area/power/delay and reliability requirements. This is similar in spirit to the idea of using high-V_{TH}/low-V_{TH}/adaptive-V_{TH} devices for reducing power consumption.

4 Conclusions

New reliability enabled EDA tools are required to accurately calculate the reliability of future nano-circuits. Current generic reliability tools are not good enough because of three main reasons:

- They cannot handle the massive size of future nano-circuits.
- They customarily assume that all gates have the same reliability, which has been recently proven to introduce too wide approximations.
- They cannot be seamlessly integrate with current EDA tools for easily trading reliability versus area/power/delay during the various design phases.

Therefore, in this paper we have introduced a methodology for an EDA tool (currently under development), that is able to accurately calculate the reliability of future nano-circuits while addressing the shortcomings of the generic reliability tools.

References

1. Hullmann, A.: Who is winning the global nanorace. Nature Nanotech. 1, 81–83 (2006)
2. Beiu, V.: A novel highly reliable low-power nano Architecture: When von Neumann augments Kolmogorov. In: IEEE International Conference on Application-Specific Systems, Architectures and Processors, Galveston, TX, USA, pp. 167–177 (2004)
3. Drexler, K.E., Randall, J., Corchnoy, S., Kawczak, A., Steve, M.L. (eds.): Productive nanosystems: A technology roadmap. Battelle Memorial & Foresight Nanotech (2007)
4. International Technology Roadmap for Semiconductors (ITRS), Semiconductor Industry Association, SEMATECH, Austin, TX, USA (2007, 2008), `http://public.itrs.net/`

5. Constantinescu, C.: Trends and challenges in VLSI circuit reliability. IEEE Micro 23, 14–19 (2003)
6. Sivakumar, P., Kistler, M., Keckler, S.W., Burger, D., Alvisi, L.: Modeling the effect of technology trends on soft error rate of combinatorial logic. In: Intl. Conf. Dependable Sys. and Networks DSN 2002, Washington, DC, USA, pp. 389–398 (2002)
7. Bhaduri, D., Shukla, S.K.: NANOLAB—A tool for evaluating reliability of defect-tolerant nanoarchitectrures. IEEE Transactions on Nanotechnology 4, 381–394 (2005)
8. Sulieman, M.H.: On the reliability of interconnected CMOS gates considering MOSFET threshold-voltage variations. In: International ICST Conference on Nano-Networks, Luzern, Switzerland (2009) (in press)
9. Elias, P.: Computation in the presence of noise. IBM Journal of Research and Development 2, 346–353 (1958)
10. Beiu, V., Aunet, S., Nyathi, J., Rydberg, R.R., Ibrahim, W.: Serial addition: Locally connected architectures. IEEE Transactions on Circuits and Systems I 54, 2564–2579 (2007)
11. Ibrahim, W., Beiu, V., Sulieman, M.H.: On the reliability of majority gates full adders. IEEE Transaction on Nanotechnology 7, 56–67 (2008)
12. Ibrahim, W., Beiu, V., Alkhawwar, Y.: On the reliability of four full adder cells. In: International Conference on Innovations in Information Technology, Dubai, UAE, pp. 720–724 (2007)
13. Beiu, V., Ibrahim, W., Lazarova-Molnar, S.: A fresh look at majority multiplexing—When devices get into the picture. In: IEEE International Conference on Nanotechnology, Hong Kong, China, pp. 883–888 (2007)
14. Beiu, V., Ibrahim, W., Lazarova-Molnar, S.: What von Neumann did not say about multiplexing: Beyond gate failures—The gory details. In: Sandoval, F., Prieto, A.G., Cabestany, J., Graña, M. (eds.) IWANN 2007. LNCS, vol. 4507, pp. 487–496. Springer, Heidelberg (2007)
15. Ibrahim, W., Beiu, V., Alkhawwar, Y.A., Sulieman, M.H.: Gate failures effectively shape multiplexing. In: IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, Washington, DC, USA, pp. 29–40 (2006)
16. Geist, R., Trivedi, K., Dugan, J.B., Smotherman, M.: Design of the hybrid automated reliability predictor. In: IEEE/AIAA Digital Avionics Systems Conference, pp. 16.5.1–16.5.8 (1983)
17. Sahner, R.A., Trivedi, K.S.: Reliability modeling using SHARPE. IEEE Trans. Reliability 13, 186–193 (1987)
18. Hirel, C., Sahner, R.A., Zang, X., Trivedi, K.S.: Reliability and performability modeling using SHARPE. In: Haverkort, B.R., Bohnenkamp, H.C., Smith, C.U. (eds.) TOOLS 2000. LNCS, vol. 1786, pp. 345–349. Springer, Heidelberg (2000)
19. Boyd, M.A., Bavuso, S.J.: Simulation modeling for long duration spacecraft control systems. In: Annual Reliability & Maintainability Symposium, Atlanta, GA, USA, pp. 106–113 (1993)
20. Couvillion, J., Freire, R., Johnson, R., Obal II, W.D., Qureshi, M.A., Rai, M., Sanders, W.H., Tvedt, J.E.: Performability modeling with UltraSAN. IEEE Software 8, 69–80 (1991)
21. Clark, G., Courtney, T., Daly, D., Deavours, D., Derisavi, S., Doyle, J.M., Sanders, W.H., Webster, P.: The Möbius modeling tool. In: International Workshop on Petri Nets and Performance Models, Aachen, Germany, pp. 241–250 (2001)
22. Ciardo, G., Jones, R.L., Miner, A.S., Siminiceanu, R.: Logical and stochastic modeling with SMART. In: Kemper, P., Sanders, W.H. (eds.) TOOLS 2003. LNCS, vol. 2794, pp. 78–97. Springer, Heidelberg (2003)

23. Dugan, J.B., Venkataraman, B., Gulati, R.: DIFTree: A software package for the analysis of dynamic fault tree models. In: Annual Reliability and Maintainability Symposium, Philadelphia, PA, USA, pp. 64–70 (1997)
24. Coppit, D., Sullivan, K.J.: Galileo: A tool built for mass-market applications. In: International Conference on Software Engineering, Limerick, Ireland, pp. 273–282 (2000)
25. Kwiatkowska, M., Norman, G., Parker, D., Segala, R.: Symbolic model checking of concurrent probabilistic systems using MTBDDs and simplex. Technical Report CSR-99-01, School of Computer Science, University of Birmingham, Birmingham, UK (1999)
26. Horton, G.: A new paradigm for the numerical simulation of stochastic Petri nets with general firing times. In: European Simulation Symposium. Verlag, Dresden (2002)
27. Lazarova-Molnar, S., Horton, G.: Proxel-based simulation for fault tree analysis. In: Symposium Simulationstechnik. SCS European Publishing House (2003)
28. Patel, K.N., Markov, I.L., Hayes, J.P.: Evaluating circuit reliability under probabilistic gate-level fault models. In: International Workshop on Logic Synthesis, Laguna Beach, CA, USA, pp. 59–64 (2003)
29. Levin, V.L.: Probability analysis of combination systems and their reliability. Engineering Cybernetics 6, 78–84 (1964)
30. Krishnaswamy, S., Viamontes, G.F., Markov, I.L., Hayes, J.P.: Accurate reliability evaluation and enhancements via probabilistic transfer matrices. In: Design Automation and Test Europe, Munich, Germany, pp. 282–287 (2005)
31. Taylor, E.R., Han, J., Fortes, J.A.B.: Towards accurate and efficient reliability modeling of nanoelectronic circuits. In: IEEE International Conference on Nanotechnology, Cincinnati, OH, USA, pp. 395–398 (2006)
32. Beg, A., Ibrahim, W.: On teaching circuit reliability. In: Annual Frontiers in Education Conference, Saratoga Springs, NY, USA, pp. T3H12–T3H17 (2008)
33. Beg, A., Ibrahim, W.: Relating reliability to circuit topology. In: IEEE North Eastern Workshop on Circuits and Systems, Toulouse, France (2009) (in press)
34. Beg, A.: Improving nano-circuit reliability estimates by using neural methods. In: International ICST Conference on Nano-Networks, Luzern, Switzerland (2009) (in press)
35. Rejimon, T., Bhanja, S.: Probabilistic error model for unreliable nano-logic gates. In: International Conference on Nanotechnology, Cincinnati, OH, USA, pp. 47–50 (2006)
36. Rejimon, T., Bhanja, S.: Time and space efficient method for accurate computation of error detection probabilities in VLSI circuits. In: IEEE Proceedings Computers and Digital Techniques, vol. 152, pp. 679–685 (2005)
37. Ibrahim, W., Beg, A., Amer, H.: A Bayesian based EDA tool for accurate VLSI reliability evaluations. In: International Conference on Innovations in Information Technology, Al-Ain, UAE, pp. 101–105 (2008)
38. Beiu, V., Ibrahim, W.: On CMOS circuit reliability from the MOSFETs and the input vectors. In: IEEE Workshop on Dependable and Secure Nanocomputing, Estoril/Lisbon, Portugal (2009)
39. Ibrahim, W., Beiu, V., Amer, H.: Why should we care about the input vectors. In: IEEE North East Workshop on Circuits and Systems, Toulouse, France (2009) (in press)