Fault-Tolerance and Noise Modelling in Nanoscale Circuit Design

Jahanzeb Anwer, Ahmad Fayyaz,
Muhammad M. Masud and Saleem F. Shaukat
Electrical Engineering Department
COMSATS Institute of Information Technology
Defense Road, Off-Raiwand Road, Lahore, Pakistan
Contact No: +60136057683
jahanzebanwer2002@gmail.com

Usman Khalid and Nor H. Hamid
Electrical & Electronics Engineering Department
Universiti Teknologi PETRONAS
Bandar Seri Iskandar, Tronoh, Perak, Malaysia
Contact No: +60192786127
hishmid@petronas.com.my

***Abstract*— Fault-tolerance in integrated circuit design has become an alarming issue for circuit designers and semiconductor industries wishing to downscale transistor dimensions to their utmost. The motivation to conduct research on fault-tolerant design is backed by the observation that the noise which was ineffective in the large-dimension circuits is expected to cause a significant downgraded performance in low-scaled transistor operation of future CMOS technology models. This paper is destined to give an overview of all the major fault-tolerance techniques and noise models proposed so far. Summing and analysing all this work, we have divided the literature into three categories and discussed their applicability in terms of proposing circuit design modifications, finding output error probability or methods proposed to achieve highly accurate simulation results.**

Introduction

The key area of research in integrated circuit design attracting many engineers and scientists is fault-tolerant computation. The need for this research arose from the need to downscale transistor dimensions for our future digital circuits (to achieve high device density). During the downscaling period, signal level has been decreasing at a fast pace whereas amount of noise is the same causing a high signal error-rate for upcoming transistor models. Therefore, we are in dire need to model noise in nanoscale circuits for simulation purposes.

The research of fault-tolerant design can be divided into three categories (based on the literature available). The first approach works on the transistor level i.e. proposing modifications in the circuit design (based on mathematical models) that can cater noise present in the circuit. The second approach is to design error-probabilistic schemes that can let the designer know how fault-tolerant his circuit is, by providing the output error probability of his circuit. The final approach proposes realistic simulation models that can take into account effect of noise dominant in nanoscale design only. The information regarding the extent to which certain transistor parameters affect the noise immunity of a circuit also comes in the third category. The common design strategy to be noticed in all these models (except for redundancy) is their dependence on probabilistic computation [1], [2]. Probabilistic analysis is used because the nature of noise is random (or probabilistic). Now, we are going to discuss the fault-tolerant design schemes and their scope followed by a general discussion on their applicability and effectiveness.

A. Redundancy

Redundancy is the basic approach to design a fault-tolerant circuit model [1]. The idea of this technique is to introduce redundancy for each gate in the circuit (or for that portion of the circuit probable of being in error) and then taking the output from the majority output decision of the original and copied gates so that if one gate in the redundant combination is faulty, the output is not affected. This technique is further divided into triple modular redundancy (TMR), cascaded triple modular redundancy (CTMR) and triple interwoven redundancy (TIR).

B. Markov Random Field(MRF) model

The most significant noise at the nanoscale level is the thermal noise. To deal with this noise, MRF equivalents [3] of universal gates have been proposed that can very well isolate the effect of thermal noise in the circuit and prevent it from affecting the final output. The final output comes out to be clean as if there were no noise in the circuit.

According to this model, a Gaussian noise source is added at the input of each logic gate. This noise source is accounted for the thermal noise generated from all the components in the previous circuit stage. In this way, the thermal noise effect of the previous stage is catered by the MRF equivalent of the current stage unless we are left with thermal noise affect of final stage of the circuit only. A more analytical model of thermal noise is rather impractical (that can model thermal noise originating from every component of a circuit); we consider the thermal noise model in this technique to be adequate for simulation purposes. We generated random noise data in MATLAB and integrated it with the VPWLF function in Cadence Analog Design Environment. By adding the VPWLF noise source at the input of every circuit stage (in the transistor-level schematic), we can model the effects of thermal noise. This noise, if added to a digital pulse input would look like as shown in Fig 1. Simulations ([3] and [4]) show that modeling thermal noise in a circuit cause many unnecessary bit reversals in simple CMOS gate as compared to almost noiseless output of MRF-CMOS gate. The drawback of this technique is the immense increase in the number of transistors required for a simple circuit. But, for improved circuit reliability, high transistor count is the price a circuit designer has to pay. (Interested readers can find MRF mathematical model in [5] and MRF-CMOS transformation method in [4]).

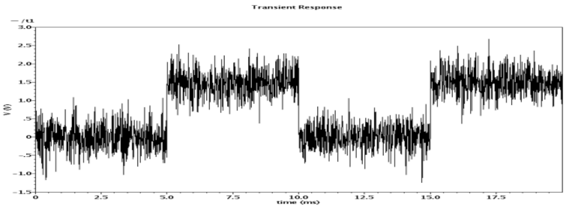

Fig 1: Thermal (Gaussian) noise added to a digital pulse (produced by simulating in Cadence)

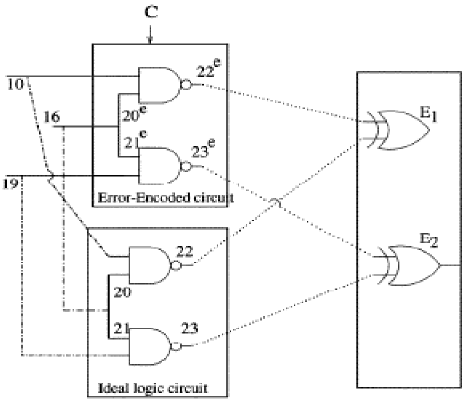

Fig 2: A conceptual idea of Bayesian Network error calculation

C. Ensemble-Dependent Matrix (EDM) Model

This model [6] also proposes modifications on the transistor level. It requires that, in the process of designing a circuit, we should consider other possible logic diagrams (having same logic equation) of a circuit stage instead of using the simplified logic stage equivalent only. Although the output of all these equivalents will be same but the bit error rate (BER) is different for each. Hence, the logic equivalent (for each circuit stage) having minimum BER should be used in the circuit so that the overall circuit could be made as fault-tolerant as possible.

The EDM model calculates the BER using the matrix method. In this study, the actual BER of the circuits is compared to the results calculated using HSPICE simulation which turned out to be almost the same. That is why, EDM is claimed to be representing actual behaviour of the circuit [6].

This study introduces another factor called ‘trace’ and shows that it is proportional to BER. Hence, the smaller the trace value, more fault-tolerant is the circuit. EDM is seemingly a good choice for CAD tools development and authors in [6] claim to introduce software model in future work. The research work in [6] also proved that EDM and MRF mathematical models converge for digital signals.

D. Bayesian Probabilistic Error Model

Bayesian networks [7] are directed acyclic graphical representations showing joint probabilities between the nodes. This model computes the error probability of the system by comparing difference between the error-free and error-encoded circuit outputs and if the there is a mismatch between the two output values; the comparator linking the two systems will output logic ‘1’. The probability of comparator output being in logic ‘1’ provides the error probability of the circuit.

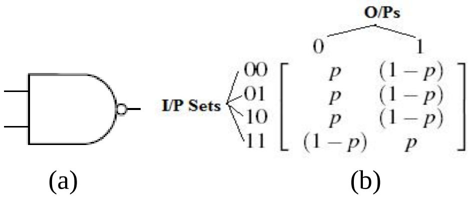

Fig 3: (a) NAND logic gate (b) NAND’s PTM

This concept has been illustrated in Fig 2 (derived from [7]) where difference in E_1 and E_2 is an indication of error.

The output-error probabilities have been calculated for exact and approximate inference schemes. The authors in [7] have designed an algorithm, logic induced probabilistic error model (LIPEM), and used software tools HUGIN and SMILE, for error-probability computation. For small benchmark circuits (e.g. LGSynth’93), exact inference scheme whereas for large benchmark circuits (e.g. ISCAS’85), approximate inference scheme has been used. Readers interested in detailed methodology and results of this method should refer to [7].

E. Probabilistic Transfer Martices (PTM) model

This model has been designed in [8], [9] to calculate output error probability of combinational circuits. It works by calculating PTM for each gate in a circuit. Here we outline the concept of PTM calculation for NAND gate using Fig 3 (reproduced through [8]). In this figure, ‘p’ denotes the ‘gate error probability’. For each input set, the output error probability is ‘p’ for an incorrect and ‘1-p’ for the correct output. Hence, the PTM is basically a matrix representing error probabilities for all node combinations in a network.

After calculating PTMs of all gates in the circuit, we divide it into stages and calculate the PTM of the entire circuit by a method that involves computing tensor products and matrix multiplications. Finally, the reliability of a circuit can be found by the formula (1). [9]

$$\text{Reliability } (v, M, J) = \| v\,(M.{*}J) \| \qquad (1)$$

where ‘v’ is the input vector, ‘M’ is the PTM of the entire circuit and ‘J’ is the identity transfer matrix, ITM. Hence, the output error probability can be found by subtracting the reliability value from unity.

F. Probabilistic Gate model (PGM)

Like PTM and Bayesian schemes, PGM [10] is based on calculating error probability of a circuit. The gate error models used in this method have been developed by Von Neumann approach and used in the PGM computation. The inputs and outputs in this scheme are considered to be independent of each other. Overall reliability (and hence output error probability) is calculated by multiplying reliabilities of each output of a circuit. Results obtained by this method have been compared with those of PTM and the output error probabilities are found to be in close comparison for both techniques.

G. Boolean Difference Error Calculator (BDEC)

The BDEC is another error probabilistic model that claims to be better in efficiency, execution time and memory usage than PTM [11]. The concept of BDEC is explained in Fig 4 [11]. According to the figure, the inputs required by the calculator are p_i (probability of i^{th} input being in logic 1), e_i (error probability of input i), f (logic equation of the gate) and e_g (gate error probability). The output e_z is calculated by a complex mathematical model involving differential equations whereas the software SIS and MATLAB have been used for simulation purposes. The research work in [11] compares BDEC with PTM and PGM, reporting close comparison of results obtained for all three techniques.

H. Flicker and Random Telegraph Signal Noise Model

The flicker noise (1/f) in MOSFETs is associated with both carrier number fluctuations and correlated carrier mobility fluctuations [12]. “Carrier number fluctuations come from the random trapping and detrapping of free carriers in the oxide traps near the Si – SiO2 interface, where the trapped carriers limit the mobility of the free carriers near the interface by Coulombic scattering” [12]. This noise is applicable to long-channel transistors and convert to random telegraph signal (RTS) noise if considered for short-channel specifically nanoscale circuits.

In [12], noise models for both NMOS and PMOS have been developed for flicker and random telegraph signal (RTS) noises. These models have been programmed in hardware description language, VerilogA and integrated into Cadence simulation software. These models automatically add flicker or RTS noise in output current of both PMOS and NMOS (based on the mathematical models of these noises). Although this model has been designed for TSMC 350nm and TSMC 35nm CMOS technologies, it can be extended to other technologies by modifying the programming code. In real operation of the circuit, the flicker or RTS noise frequency is not dominant but for simulation purposes, the designers add high frequency of this noise to provide reliable simulation results for future predictive CMOS technology models.

We have simulated noisy-NMOS model in Cadence Analog Design Environment and the results are shown in Fig 5. This figure shows the noise addition in current (which automatically gets reflected in the output voltage). By using this realistic MOSFET model in simulations, the designer would come to know the limitations that could arise due to this noise.

I. Poisson Noise Model

The Poisson noise model has been developed ([13]) to help computing the soft-error rate (SER) of a circuit due to thermal noise. In this work, noise characteristics of a circuit have been modelled for 65nm CMOS technology operating in sub-threshold region (V_{dd}=0.2V). The analysis in this model is based on the birth-death queue model which is based on charging load capacitance of the logic circuit. For the case of inverter, the noise has been modelled by variation of load capacitor charge. The variation in this charge (proportional to the output current) is simulated against time samples. This variation is clearly in agreement with the Gaussian noise model we used for thermal noise equivalent (in MRF). Using this analysis scheme, mean time to first error has been calculated under different bias and threshold voltages.

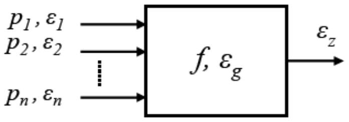

Fig 4: Block diagram of BDEC

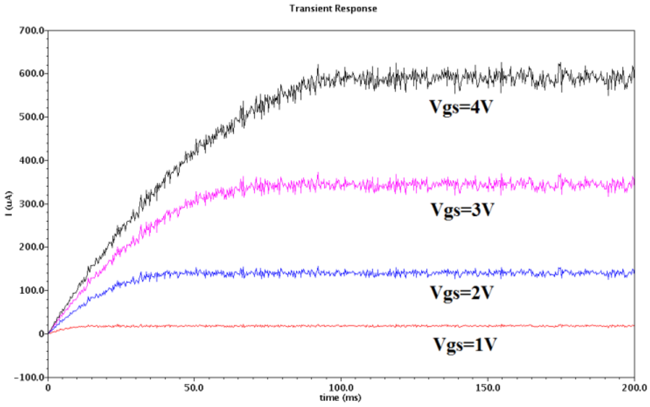

Fig 5: Variation of V_{DS} with V_{GS}

J. Plackett and Burman Screening Method

Statistical screening methods like plackett and burman has been used in [14] to provide information on the effect of different parameter variations (of transistor) on SER estimation. It is a very useful research conducted to help circuit designers minimize the SER in their designs. It provides information on how and to what extent certain transistor parameters affect SER. Simulation results show that the most critical parameters in the design process are supply voltage and transistor technology (model) followed by transistor with, injection current model, threshold voltage and fanout in order of decreasing significance. For detailed simulation results on how does these parameters actually affect the circuit, refer to [14].

Discussion

The first category contains techniques that focus architecture (transistor) level changes in the circuit design. This category consists of redundancy, MRF and EDM. Redundancy, although still in use, is the oldest fault-tolerance scheme which is costly both in terms of area and power consumption. The voter in redundant architecture has a complex design and if it fails, the whole architecture collapses. It handles manufacturing errors. MRF on the other hand handles transient errors. It is an elegant technique which has a noise-immunity far better than any scheme. The shortcoming of this design is the immense increase in the number of transistors required. For example, the MRF-CMOS inverter requires 34 transistors in contrast to the 2 transistors required for a simple CMOS inverter. Another disadvantage is the difficulty to model this technique into software tools due to its complex implementation model and need to divert from system-level perspective to component-level for cost-effective system design [4]. EDM, although looks more promising solution in terms of transistor count offers no automatic simulation method of identifying circuit alternative with minimum BER. EDM though claims to be in progress of integrating its methodology in software in future, which if happens, could possibly replace MRF as the best architecture level solution.

The second category consists of techniques that compute error-probability of a circuit. If we look at the circuit simulators of today, they show us memory usage, power consumption, processing time, etc but none of the software has an error-probability calculator. Output error probability is the direct indication of fault-tolerance capability of a circuit. But before integrating error-tolerance techniques into software, we should have their firm mathematical models available. This category consists of probabilistic error models, Bayesian, PTM, PGM and BDEC. All of these techniques show their mathematical models and simulation results (in the research publications) but none of them have been used as an error probability calculation tool in software. We are going to compare these four techniques on the basis of results obtained for a common circuit simulated in all these techniques i.e. C17 benchmark circuit. The values of output error probability (for individual gate error probability= 0.05) calculated are 0.1342, 0.216, 0.238 and 0.234 for Bayesian, PTM, PGM and BDEC respectively. We can observe that the Bayesian output is the only odd result whereas the other three methods have comparable results. So, in the first glance, we can assume that Bayesian model lacks in some thorough computation; a possible reason for its result in strong contradiction with other methods but since we cannot have a validation standard, we cannot pass a hard comment on Bayesian modeling. In terms of execution time comparison, PGM though shows no results whereas BDEC is a timing efficient method is orders of magnitude than PTM. Bayesian accounts for zero execution time which is because its time-scale is in seconds whereas PTM and BDEC have a milliseconds scale. An important observation in all error-probabilistic analysis is that the significance of gate error probability has been described in none of its related literature. Instead, their arbitrary values have been assumed for all four techniques described earlier.

The last category consists of research-work that aids in achieving more practical simulation results before implementation of our circuits. Flicker noise model has been integrated in the software Cadence and is no-doubt helpful in achieving reliable simulation results. Although flicker (or RTS noise) has a small magnitude and low frequency, it can be a source of errors in future deep sub-micron devices due to their downscaled dimensions and low supply voltages. Poisson model is used to simulate the effects of threshold voltage variation and transistor width on the SER, which can be useful if implemented in software analysis as well in future. Plackett and Burman screening method provides a thorough analysis of many factors that actually affect the noise immunity of a system and is a detailed affects' report uncovered previously by any of the researchers.

Conclusions

For architecture level fault-tolerance solutions, MRF is the only suitable approach for now. Error-probabilistic schemes are all on the same standing with possible exception of Bayesian. Simulation contribution of flicker noise model is significant whereas Poisson model and Plackett and Burman screening method are a source of critical information for circuit designers. We hope that this comparative study provides a broad view of fault-tolerance techniques to researchers striving for high-speed and error-free nano-computation.

References

[1] Sumit Ahuja, Gaurav Singh, Debayan Bhaduri and Sandeep K. Shukla, “Fault and Defect Tolerant Architectures for Nano-computing,” in Bio-Inspired and Nanoscale Integrated Computing, Mary Eshaghian-Wilner, Wiley, 2009.

[2] R. I. Bahar, J. Chen and J. Mundy, “A Probabilistic-Based Design for Nanoscale Computation,” in Nano, Quantum and Molecular Computing: Implications to High Level Design and Validation, Sandeep K. Shukla and R.I. Bahar, Springer, 2004.

[3] Kundan Nepal, “Markov Random Field,” in Designing Reliable Nanoscale Circuits Using Principles of Markov Random Fields, PhD dissertation, Brown University, RI 02912, US, 2007.

[4] Jahanzeb Anwer, Usman Khalid, Narinderjit Singh, Nor H. Hamid, Vijanth S. Asirvadam, “Highly Noise-Tolerant Design of Digital Logic Gates using Markov Random Field Modelling,” to be published in 2nd International Conference on Electronic Computer Technology (ICECT), Kuala Lampur, Malaysia, May 2010.

[5] Jahanzeb Anwer, Usman Khalid, Narinderjit Singh, Nor H. Hamid, Vijanth S. Asirvadam, “Joint and Marginal Probability Analyses of Markov Random Field Networks for Digital Logic Circuits,” to be published in 3rd International Conference on Intelligent and Advanced Systems (ICIAS), Kuala Lampur, Malaysia, June 2010

[6] Huifei Rao, Jie Chen, Changhong Yu, Woon Tiong Ang, I-Chyn Wey, An-Yeu Wu and Hong Zhao,” Ensemble Dependent Matrix Methodology for Probabilistic-Based Fault-tolerant Nanoscale Circuit Design,” in IEEE International Symposium on Circuits and Systems (ISCAS), New Orleans, LA, USA, pp. 1803-1806, May 2007.

[7] Thara Rejimon, Karthikeyan Lingasubramanian, and Sanjukta Bhanja, “Probabilistic Error Modeling forNano-Domain Logic Circuits.” in IEEE Transactions on Very Large Scale Integration (VLSI) Systems. Vol. 17, No. 1, CA, USA, Jan 2009.

[8] S. Krishnaswamy, G.F. Viamontes, I.L. Markov, J.P. Hayes, “Accurate reliability evaluation and enhancement via probabilistic transfer matrices,” in Proc. Design, Automation and Test in Europe, pp: 282-287. 2005.

[9] S. Krishnaswamy, G.F. Viamontes, I.L. Markov, J.P. Hayes, “Probabilistic transfer matrices in symbolic reliability analysis of logic circuits,” in ACM Transactions on Design Automation of Electronic Systems (TODAES), Volume 13 , Issue 1, Article No.: 8, Jan 2008.

[10] Jie Han, Erin Taylor, Jianbo Gao and José Fortes, “Faults, Error Bounds and Reliability of Nanoelectronic Circuits,” in Proc. 16th International Conference on Application-Specific Systems, Architecture and Processors (ASAP’05), pp. 247-253, 2005.

[11] Nasir Mohyuddin, Ehsan Pakbaznia and Massoud Pedram, “Probabilistic error propagation in logic circuits using the Boolean difference calculus.” in Proc. 26th International Conference on Computer Design, ICCD, pp. 7-13, CA, USA, 2008.

[12] Nor Hisham hamid, “Modelling Noisy MOSFETs,” in Can deep-sub-micron device noise be used as the basis for probabilistic neural computation, PhD dissertation, The University of Edinburgh, 2006.

[13] H. Li, J. Mundy, W.R. Patterson, D. Kazazis, A. Zaslavsky, and R.I. Bahar, "A Model for Soft Errors in the Subthreshold Cmos Inverter," in Proceedings of Workshop on System Effects of Logic Soft Errors, Nov. 2006.

[14] Drew C. Ness and David J. Lilja, “Guiding Circuit Level Fault-Tolerance Design with Statistical Methods,” in Proc. Design, automation and test in Europe, pp.348-353, Munich, Germany, 2008.