Reliable circuit analysis and design using nanoscale devices

Renu Kumawat*, Vineet Sahula*, M..S. Gaur[a]
[*]Dept. of Electronics and Communication Engg., [a]Dept. of Computer Engineering,
Malviya National Institute of Technology, Jaipur, India - 302017

ABSTRACT

The miniaturization of the devices into nanoscale has enabled ultra high density chips, but at the cost of increased defect density. In this manuscript, Markov Random Field (MRF) approach is used to evaluate the device reliability in the presence of high defect density. Both hard and soft errors have been considered. We have presented a NANOLAB based fault model of 8-bit full adder, basic building block being 2:1 multiplexer. At each level, a Triple Modular Redundancy (TMR) is employed to enhance reliability. The results are compared with another 8-bit full adder, designed using logic gates. Assuming defect rate up to 10%, the circuits are evaluated for stuck at faults. Further, we have augmented the NANOLAB tool to include a design library of various types of flip flops. A 4-bit SISO right shift register is used as vehicle for exemplifying our approach. The fault tolerant approach N-Modular Redundancy (NMR) is compared at different levels of granularity and for varying levels of N. It is observed that NMR fails to provide the device fault tolerance when defect rate is higher than a threshold value.

Keywords: Markov Random Field, N Modular Redundancy, reliability, defect rate, NANOLAB, stuck-at-faults, hard errors and soft errors.

1. INTRODUCTION

Conventional silicon technology will reach some fundamental limits in the next few years. There are unavoidable physical and financial obstacles in this continual downsizing of silicon to nanoscale. The miniaturization of the device causes hard and soft errors to be inevitable. Thus, we achieve the ultra high density chips, but at the cost of high defect rate. The soft errors are due to noise and thermal perturbation while hard errors are due to fabrication defects and process variation. The energy difference between logic states will approach a thermal limit. Hence, the device becomes inherently unreliable and contingent. The stochastic nature of the device helps to model it as a Markov Random Field (MRF). This probabilistic modeling of the devices at nanoscale, assumes logic states to be random variables which interact through a distribution representing their joint probability. Under this framework, each logic variable is represented as a node and statistical dependence between these variables is modeled as an edge of MRF Graph. Clique energy function is derived from the graph or this function can be obtained as summation over all valid states in logic compatibility table. Each variable has some random probability of having state '0' or '1'. At the primary output, joint probability is obtained by summation over the marginal probabilities of intermediate nodes. The belief propagation algorithm[1] has been used to calculate marginal probabilities. To increase the system reliability in presence of high defect rate, some defect tolerance techniques must be incorporated. The defect tolerance includes detecting the fault type, locating it and then avoiding these faults by redundancy or reconfiguration types of techniques. Literatures are available on N-Modular redundancy (NMR), Cascaded Triple Modular Redundancy (CTMR), NAND-Multiplexing, etc[2,3]. In all these techniques, a single design entity is replaced by multiple entities to enhance system reliability on the cost of increased hardware. All these approaches become impractical when defect rate is as high as 10^{-3}, since the redundancy will increase exponentially. To evaluate this reliability vs. redundancy trade off, a MATLAB based tool called NANOLAB[4,5] has been developed. NANOLAB is a probabilistic model based on Markov Random Field and belief propagation algorithm[1,6].

The paper is organized as follows. In Section 2, we present the recent efforts in literature towards implementation and evaluation of nanoscale logic circuits. We also include a brief discussion on Markov Random Field and belief propagation algorithm. Our proposal for reliable evaluation of combinational and sequential components e.g. 8-bit ripple carry adder and a Serial-In-Serial-Out right shift register is discussed in Section 3. An N-Modular Redundancy (NMR) is compared for different levels of granularity of circuit/logic and for varying levels of N. We conclude in Section 4.

International Conference on Communication and Electronics System Design,
edited by M. Salim, K. K. Sharma, V. Janyani, Proc. of SPIE Vol. 8760, 87602C
 · doi: 10.1117/12.2012516

2. BACKGROUND

The congregation of nanoelectronic devices can be modeled as a Markov Random Field, where a probabilistic logic value can be associated with each primary input node of the device. The marginal probabilities of occurrence of logic '0' or '1' at other intermediate nodes in such a Boolean network are computed using belief propagation algorithm. Such a probabilistic methodology of circuit designing is provided by NANOLAB[4,5] tool.

2.1 Markov Random Field

Markov Random Field[6] (MRF) can be considered as a model in which a set of random variables having Markov property are described through an undirected graph. Let us consider a set of random variables $\chi = \chi_1, \chi_2, \ldots, \chi_n$. A state value x_i is associated with the corresponding random variable χ_i. The Markov random process has Markov property which states that if conditional probability distribution of future states of the process, given the present state and a few of past states, depends only on present state and is independent of the past states. This definition of Markov property can be extended to MRF. A random variable χ_i has Markov property iff it is conditionally dependent only on its neighbors. Now, let us define a Markov blanket of χ_i as η_i, such that $\eta_i \subset \chi$ and each $\chi_k \in \eta_i$ is the neighbor of χ_i. Let us assume the Positivity condition, as stated in (1) and the condition for existence of Markovinity is as given in (2).

$$P(\chi_i = x_i) > 0, \qquad \forall\ \chi_i \in \chi \tag{1}$$

$$P(\chi_i = x_i \mid \chi - \chi_i) = P(\chi_i \mid \eta_i) \tag{2}$$

Thus, as obvious from (1) and (2), MRF has positive mass or density and is pair wise Markov. In other words, MRF possess local and global Markov property. The probability distribution of MRF can be written as in (3) and is known as **Gibbs Distribution**.

$$P(\chi_i = x_i \mid \eta_i) = \frac{1}{Z} e^{\left(-\frac{1}{K_B T} E(\eta_i)\right)} \tag{3}$$

$$P(\chi_1, \chi_2, \ldots, \chi_n) = \prod_{i=1\ldots n} P(\chi_i \mid \eta_i) \tag{4}$$

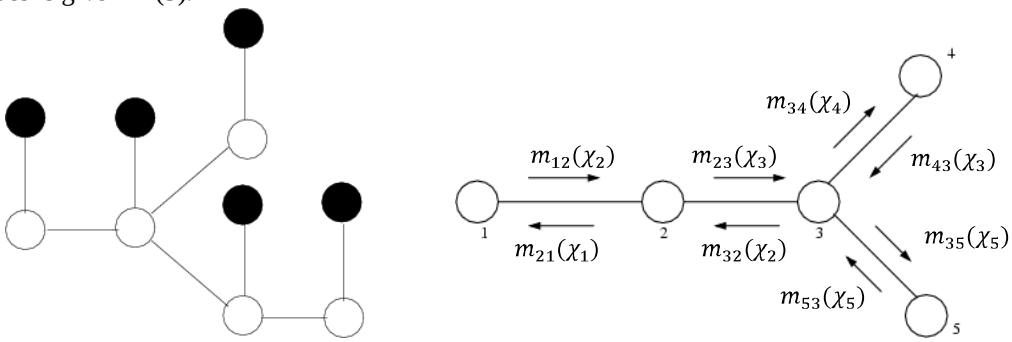
where K_B is Boltzmann constant, T is temperature, Z is normalization constant known as partition function and $E(\eta_i)$ is called clique energy function. Here, equation (3) can be derived from Hamersley-Clifford Theorem[7]. The Hamersley-Clifford Theorem can be stated as a probability distribution with positive mass or density, satisfies the local or global Markov Property with respect to an undirected graph G(V,E), iff it is a Gibbs Distribution, i.e its density can be factorized over the cliques of the Graph G. Thus, the overall joint probability of MRF is given by (4).

Nanoscale devices can be modeled as MRF. Each logic variable is represented as a node and statistical dependence between these variables is modeled as an edge of MRF Graph. Clique energy function is derived from the graph or this function can be obtained as summation of all valid states in logic compatibility table. Each variable has some random probability of occurrence of logic zero and one. At the primary output, the joint probability is obtained and for each intermediate node, the marginal probability is calculated by using belief propagation algorithm[1]. Bhaduri and Shukla[4,5] designed a MATLAB based tool called NANOLAB which is based on MRF theory for probabilistic computation at nanoscale. The NANOLAB deals with unreliable nature of nanoscale devices using N-Modular Redundancy (NMR) and NAND Multiplexing[3]. Reliability of defect tolerant circuits was evaluated in presence of noise. Nepal and Bahar[8-10] used the MRF approach to implement CMOS based devices at nanoscale. The MRF based logic gates at 70nm were designed and device characteristics were compared with CMOS based logic gates, in presence of noise and soft errors[10]. The reliability is thus achieved at the cost of increased hardware.

2.2 Belief Propagation Algorithm

The overall joint probability of MRF can be calculated by (4) but to calculate marginal probabilities at each intermediate node of graph G, we can use belief propagation algorithm[1]. In this algorithm, the observed and hidden variables of a Markov Random Field (MRF) are defined. Let us assume that there is a random variable $\gamma_i \in \chi$ called observed variable, whose parameters are already observed and a hidden variable $\chi_i \in \chi$ whose parameters are yet to be extracted. Also, let us

consider another hidden variable χ_j in neighborhood of χ_i. There is a statistical dependence between these variables, defined by a joint compatibility function $\phi(\chi_i,\gamma_i)$ or simply $\phi(\chi_i)$ and compatibility function $\phi(\chi_i\chi_j)$. An example illustration for an MRF is given in Figure 1(a), where black dots represent the hidden nodes and white nodes represent the observed nodes. The belief propagation algorithm[1], gives exact marginal probabilities for all nodes in any singly-connected network. The belief b_i at node i is proportional to local evidence at that node $\phi(\chi_i)$, and all messages $m_{ji}(\chi_i)$ coming into nodes is given in (5).



(a) (b)

Figure 1: Example network for belief propagation algorithm (a) black dots represent the hidden nodes and white nodes represent the observed nodes (b) message passing between hidden nodes[1]

$$b_i = k\,\phi_i(\chi_i)\prod_{j\,\epsilon\,\eta_i} m_{ji}\,(\chi_i) \tag{5}$$

$$m_{ij} \leftarrow \sum_{x_i} \phi_i\,(\chi_i)\psi(\chi_i,\chi_j)\prod_{j\,\epsilon\,\eta_{i-j}} m_{ki}\chi_i \tag{6}$$

The variable $m_{ji}(\chi_i)$ is nothing but a message from a hidden node j to hidden node i and k is a normalization constant such that beliefs must sum to 1. The message update rule is given in (6). This message update rule when applied recurrently generates required equation for belief as in (5). The Figure 1(b) gives an example illustration of message passing between hidden nodes of a MRF.

2.3 NANOLAB

NANOLAB is a reliability evaluation tool[4,5] that consists of a library of MATLAB functions. These functions are based on probabilistic design methodology of MRF[6]. In other words, a Boolean network is modeled as MRF such that each node has some random probability of occurrence of logic '0' or '1'. The tool is used to design nanoscale digital circuits composed of at most three input logic gates. NANOLAB takes in as input a logic compatibility table and primary input node's probabilities of having logic '0' or '1'. The in-built functions are defined to compute Gibbs energy distribution, entropy, probabilities at output of each logic gate for different K_BT values, etc. as explained in Bahar *et. al.*[6]. The belief propagation algorithm[1] is used to propagate these probabilities to intermediate nodes of the network. In NANOLAB, functions are provided to model Gaussian and uniform noise. This tool facilitates to include Triple Modular Redundancy (TMR), Cascaded Triple Modular Redundancy (CTMR) and NAND multiplexing in combinational circuits.

3. PROPOSED WORK FOR RELIABLILITY ANALYSIS

The NANOLAB tool is explored to design TMR and CTMR based logic gates. It is used to study the effects on device output in presence of noise. The NANOLAB tool is augmented to include 2:1 multiplexer as a basic building block. Thus, a design library is developed to model multiplexer based logic gates and for this necessary changes are made to the logic compatibility table. In order to enhance the system reliability, each multiplexer based logic gate is modeled by Triple Modular Redundancy (TMR). Various flip flops are modeled using logic gates and a design library is created. A 4-bit Serial-In-Serial-Out (SISO) right shift register is used as a vehicle to exemplify our approach. The combinational circuits designed using MRF based approach, are analyzed for fixed redundancy (TMR) at different levels of abstraction while considering fixed defect rate. A 4-bit ripple carry adder having TMR at each Module (TMR_M) and at each logic gate level (TMR_G) are compared to an adder without redundancy. Secondly, with varying levels of redundancy (No MR,

TMR, 5MR, 7MR, 9MR) the adder is analyzed for varying defect rates. In this case, hardware redundancy is applied to each full adder.

3.1 Combinational circuit design using 2:1 multiplexer based logic gates

We consider a design of an 8-bit ripple carry adder using mux-based logic gates. The TMR is provided at each logic gate level. As discussed earlier, necessary changes are made to logic compatibility table of gates to model them using a multiplexer. The logic compatibility table of 2:1 multiplexer based AND gate and XOR gate is given in Table 1 and Table 2, respectively. These multiplexer based logic gates are giving same equations for Gibbs Energy and same entropy values as in case of basic logic gates approach. All multiplexer based logic gates, except XOR gate, offer the advantage that one of their input lines can be directly connected to Vdd or Gnd. Hence, their logic compatibility table is reduced by half which decreases the search time for valid states during simulation.

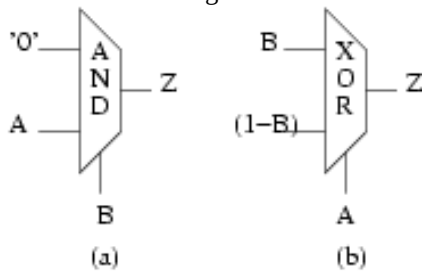


Figure 2. A 2:1 multiplexer based logic gate design (a) AND gate (b) XOR gate

Table 1. Logic compatibility table for 2:1 mux-based AND gate

Input 0	Input A	Select B	Output Z	Valid/Invalid (1/0)
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	1	1

Table 2. Logic compatibility table for 2:1 mux-based XOR gate

Input B	Input 1-B	Select A	Output Z	Valid/Invalid (1/0)
0	0	0	0	1
0	0	1	0	0
0	1	0	0	1
0	1	1	1	0
1	0	0	1	1
1	0	1	0	0
1	1	0	1	1
1	1	1	1	0

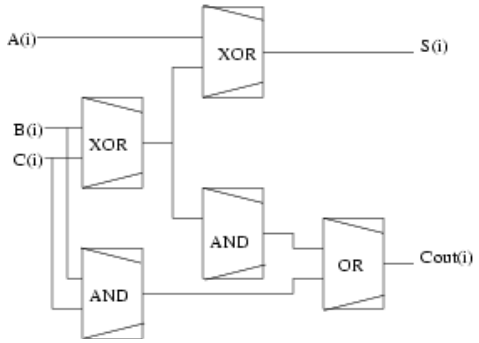


Figure 3. A 1-bit Full Adder with TMR at each mux-based logic gate

The 1-bit full adder, as shown in Figure 3, has triple modular redundancy at each level logic gate level. Using this 1-bit full adder an 8-bit ripple carry adder is designed and it is compared to an 8-bit ripple carry adder using simple logic gates. Both adders have TMR at each logic gate. The two adders are run for different input values and each time sum and carry values are same up to 2 places after decimal. As depicted from Table 3, the mux-based circuits show higher probabilities for logic '0'. When input bit-vector A = "11101010" and B = "11011111" with input carry = '1', the expected output sum is "11001001". In such a case, the probabilities of sum for with and without multiplexer based 8-bit ripple carry adders are given in Table 3. At input side, logic '1' is represented by probability vector [0.2 0.8] and logic '0' by [0.8 0.2]. The similar results can be obtained for other circuits also.

Table 3. Probabilities of getting '1' at sum output for mux-based and without mux-based 8-bit ripple carry adder

Adder output	With Mux-based Adder	With logic gate adder	Logic Value
Sum(7)	0.738954	0.739023	1
Sum(6)	0.682778	0.682954	1
Sum(5)	0.284839	0.284728	0
Sum(4)	0.260738	0.260673	0
Sum(3)	0.686478	0.686591	1
Sum(2)	0.282181	0.282115	0
Sum(1)	0.172751	0.172673	0
Sum(0)	0.724022	0.724851	1

3.2 Sequential circuit design

We designed a MATLAB library of all flip flops and latches based on MRF approach. The results of a NOR based SR flip flop are shown here as an example. The probabilities of Set and Reset are taken as an input and probability of next state output is generated. When p(set = 1) = 0.9 and p(reset = 1) = 0.1 the plot in Figure 4(a) is generated and next state probability for logic '1' is estimated as 0.8235 which can be accepted as logic '1'. Thus, flip flop is in set state. Similarly, Figure 4(b) shows reset conditions and next state probability for logic zero is 0.8353. When p(set = 1) = 0.1 and p(reset = 1) = 0.1 and present state probability for logic '0' as 0.9, the next state probability is calculated as 0.8353, as shown in Figure 4(c). This can be accepted as logic '0' and we get no change in state as desired by given input conditions. Now, these flip flops and latches can be used to design sequential circuits. We here discuss a 4-bit Serial-In-Serial-Out right shift register to exemplify our approach. A D flip flop is used as a memory element in the design of SISO. We assume that there is 90% probability of occurrence of logic '0' at the input S_{in}, then this data is shifted out serially and obtained at S_{out} after a delay of 4 flip flops. The plots generated at each flip flop from FF0 to FF3 are shown in Figure 5.

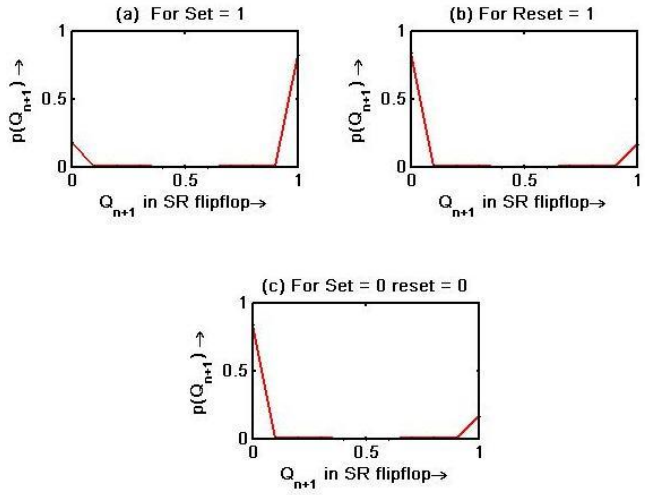


Figure 4. Simulation results: Probability of getting correct output for a NOR based SR flip flop

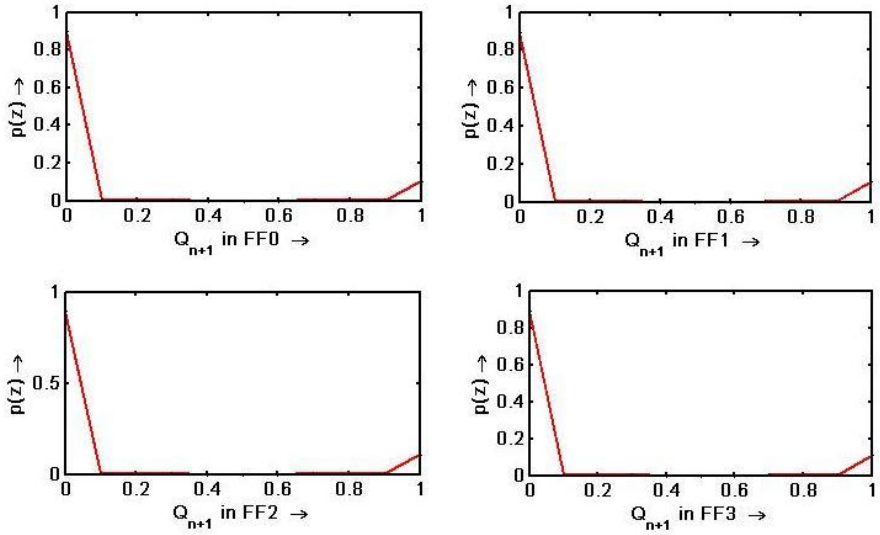


Figure 5. Simulation results: Probability of next state Q_{n+1} for a 4-bit Serial-In-Serial-Out right shift register when p (S_{in}=0) = 0.9

3.3 Reliability enhancement through hardware redundancy

3.3.1 Fixed Modular Redundancy v/s Fixed Defect Rate

Three different MRF based 4-bit Ripple Carry Adders have been designed. These adders have TMR at varying levels of granularity. Although adder1 has no redundancy, adder2 and adder3 have been designed with TMR at each 1-bit Full adder and TMR at each logic gate, respectively. The probability that correct input appears on primary input is assumed to be 0.7. The transient faults are injected randomly at any potential fault site. For fixed redundancy, various combinations of inputs are applied and faults are injected randomly. Total numbers of simulations are 50. It is observed that for some input combinations and fault sites, TMR_M is less reliable than TMR_G and vice versa. For varying redundancy, let us consider inputs to the adder be A = "0011" and B = "0110" with input carry C_{in} = '1'. The probability of logic '1' occurring on output is plotted with increasing faults in Figure 6. As expected, TMR increases the circuit's reliability. However, as depicted from Figure 6, for most of the input combinations and fault sites, TMR at module level (adder2) is less reliable than TMR at gate level (adder3).

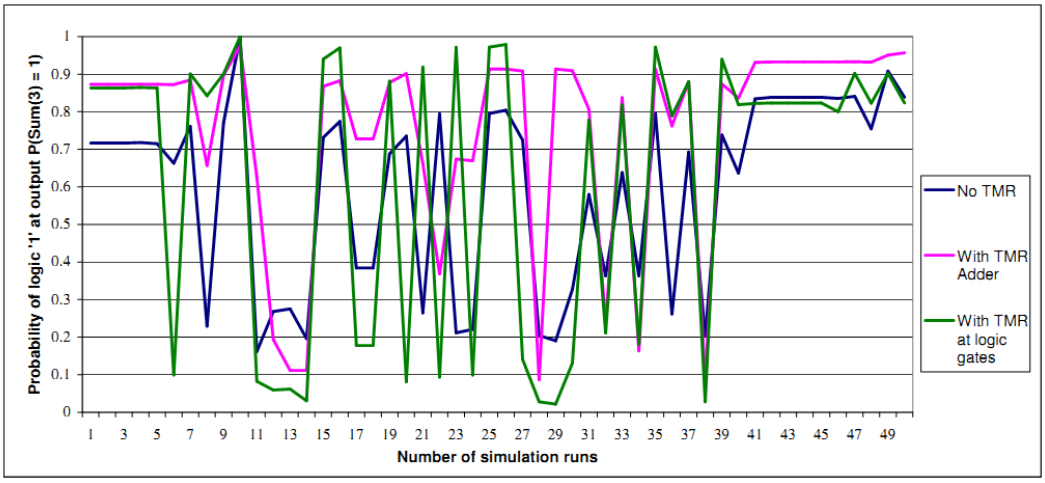


Figure 6. Probability of obtaining logic '1' on Sum(3) for a 4-bit ripple carry adder for varying defect rate Total test cases are 50 and simulation is done for (a) No redundancy (b) TMR at each 1-bit adder (c) TMR at each logic gate

3.3.2 Varying Modular Redundancy v/s Increasing Defect Rate

Here, four different MRF based 4-bit Ripple Carry Adders have been designed, namely $adder_{0mr}$, $adder_{3mr}$, $adder_{5mr}$ and $adder_{7mr}$. The modular redundancy has been introduced at gate level. This means $adder_{imr}$ has been designed with modular redundancy i at each logic gate, for i ={ 0, 3, 5,7}. The defects have been introduced randomly and increased by one at each simulation run. Let us consider inputs to these adders be A = 0011 and B = 0110 with input carry C_{in} = 1. The probability of logic '1' occurring on output has been plotted with increasing faults in Figure 7. It has been observed that, although the reliability increases with modular redundancy, sometimes we get wrong results with this approach.

4. CONCLUSION AND FUTURE WORK

The NANOLAB tool has been used for probabilistic modeling of combinational circuits using MRF. Both hard errors and soft transient errors, with time varying defect probability, have been assumed to be present during simulation. The NANOLAB tool is augmented to include multiplexer based logic gate and flip flop library. These design entities can be directly used for combinational and sequential circuit design. Instead of 2:1 multiplexer, 4:1 multiplexer or bigger ones can be used as basic building block. The fault tolerant approach, N-Modular Redundancy (NMR) has been compared at different levels of granularity and for varying levels of redundancy. It is observed that NMR fails to make the device fault tolerant when defect rate is much higher than threshold value. We thus conclude that, the traditional fault tolerant techniques fail to provide reliability under high defect rates, at nanoscale. Although nanoscale devices possess ultra high density, so hardware is not an issue. Self reliable and self resilient circuits are required to be designed to withstand the high defect rate at nanoscale. The present work is to be extended to evaluate circuit reliability for different types of faults in presence of noise and thermal perturbation.

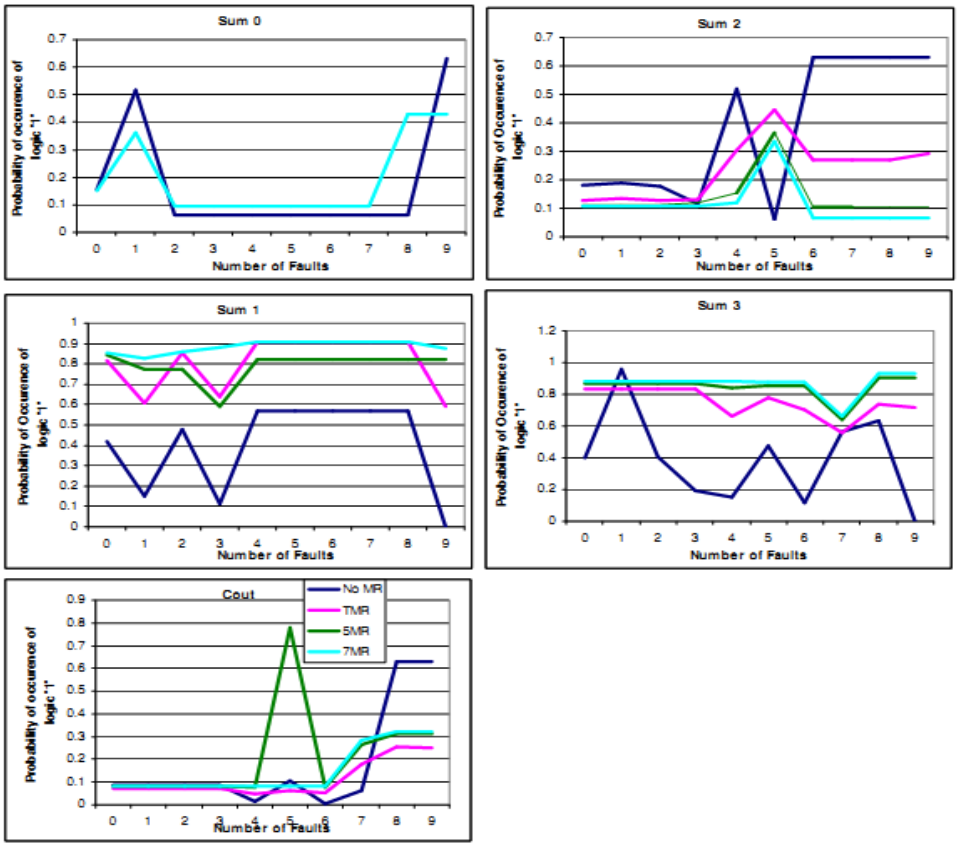


Figure 7. Probability of logic '1' at Sum(0), Sum(1), Sum(2), Sum(3) and Cout with increasing fault rate and varying levels of modular redundancy. Plots are for cases: (i) no redundancy, (ii) TMR (iii) 5MR (iv) 7MR

Acknowledgment

We gratefully acknowledge the financial support from the Ministry of Communication and Information Technology, India, provided under the sponsored project "Special Manpower Development Project for VLSI Design and related software", phase II.

REFERENCES

[1] Yedidia, J.S., Freeman, W. T. and Weiss, Y., "Understanding belief propagation and its generalizations," Exploring artificial intelligence in the new millennium, Morgan Kaufmann Publishers Inc., 239-269 (2003).

[2] Neumann, J.V., "Probabilistic logics the synthesis of reliable organisms from unreliable components," Automata Studies, , 43-98 (1956).

[3] Han, J., Gao, J., Qi, Y., Jonker, P. and Fortes, J. A. B., "Toward hardware-redundant, fault-tolerant logic for nanoelectronics," IEEE Des. Test, 22(4), 328-339 (2005).

[4] Ha, D. S., Member, B. R., and Bhaduri, D., "Tools and techniques for evaluating reliability trade-offs for nano-architectures," Nano Quantum and Molecular Computing: Implications to High Level Design and Validation" Kluwer Academic Publishers, (2004).

[5] Bhaduri, D. and Shukla, S., "Nanolab - a tool for evaluating reliability of defect-tolerant nanoarchitectures," IEEE Transaction on Nanotechnology, 4(4), 381-394 (2005).

[6] Bahar, R.I., Mundy, J. and Chen, J. "A probabilistic-based design methodology for nanoscale computation," International Conference on Computer-Aided Design, 480-486 (2003).

[7] Besag, J., "Spatial interaction and the statistical analysis of lattice systems," Journal of the Royal Statistical Society. Series B (Methodological), 36(2), 192-236 (1974).

[8] Nepal, K., Bahar, R.I., Mundy, J., Patterson, W.R. and A. Zaslavsky, A., "Designing logic circuits for probabilistic computation in the presence of noise," DAC '05: Proceedings of the 42nd annual Design Automation Conference. New York, NY, USA: ACM, 485.490 (2005).

[9] Nepal, K., Bahar, R.I., Mundy, J., Patterson, W.R. and A. Zaslavsky, A., "Designing mrf based error correcting circuits for memory elements," DATE '06: Proceedings of the conference on Design, automation and test in Europe. 3001 Leuven, Belgium, Belgium: European Design and Automation Association, 792-793 (2006).

[10] Nepal, K., Bahar, R.I., Mundy, J., Patterson, W.R. and A. Zaslavsky, A., "Designing nanoscale logic circuits based on markov random fields," J. Electron. Test. 23(2-3), 255-266 (2007).