Probabilistic Modeling Approaches for Nanoscale Devices

***Abstract*—The nanoscale circuits are highly unreliable and posses high defect density. The deterministic approaches are not applicable to such circuits. Many approaches have been proposed for probabilistic analysis and reliable design for nanoscale circuits. These includes Markov Random Field, Bayesian Network, Probabilistic Decision Diagrams, Decision Trees, Factor graphs, Tanner Graphs, Probabilistic Transfer Matrices, Influence diagrams, etc. In this paper, a comparative study of some of these approaches is presented. The Markov random field, Probabilistic Transfer Matrices and Probabilistic Decision Diagram based approaches are implemented in MATLAB and logic circuits are designed. It is observed that Bayesian Network and Probabilistic Decision Diagrams have least time complexity among these approaches. The Probabilistic Transfer Matrices require alot of memory and long simulation time.**

I. Introduction

As indicated by Moore's Law, lots of practical and economical barriers are limiting further scaling down of CMOS based semiconductor technology. Novel nano-electronic devices like NanoFabric, carbon nanotube, quantum dot, etc. are under intense research to replace present semiconductor devices. The nanoelectronic devices posses high density, require low power and offer lower manufacturing cost. However, they are inherently defective and unreliable. The imprecision due to chemical self-assembly makes the nano-devices defective and unreliable. Such defects that occur at fabrication time are called structural faults. Defects can be introduced later due to noise, radiation, gate switching, cosmic rays, thermal perturbation, etc., and are known as soft transient errors (or soft errors). These soft transient errors can occur any where and at any time in the circuit. Thus, these soft errors are hard to detect by traditional testing methodologies. Further, at nanoscale, the energy difference between logic states will approach a thermal limit [2]. As a result, logic $'0'$ (or $'1'$) at any node of nanodevice has some random probability of occurrence. This stochastic nature of the nano device shows existence of Markov property between different nodes of devices. The deterministic approaches fail to model such random behavior and thus lots of research is going on for probabilistic modeling of nanoscale devices [1]–[5], [7], [8], [10]–[16]. These probabilistic approaches assume that there is certain probability of occurrence of logic $'0'$ (or $'1'$) on a signal (wire). Reliable computation via redundancy trade off might be an alternative for modeling nanoscale circuits. But many of such proposed approaches assume reliable voter which is not practical in presence of soft error.

In this manuscript, we have presented a comparative study of different probabilistic approaches for modelling nano devices. Brief overview of probabilistic modeling approaches like Markov Random Field, Bayesian Network, Probabilistic Transfer Matrix and Probabilistic Descision Diagrams is presented. We have designed basic logic gates using these approaches in MATLAB. With these logic gates, combinational circuits are designed and compared in terms of performance. The remainder of this paper is organized as follows. In Section II, we present different probabilistic approaches for reliable circuit design at nanoscale in literatures. This includes a brief discussion on approaches like Markov Random Field (MRF), Bayesian Networks(BN), Probabilistic Decision Diagram(PDD) and Probabilistic Transfer Matrices (PTM). The Section III explains the combinational circuits designed and implemented with these approaches. We finally conclude with future work in Section IV.

II. Probabilistic Approaches for modeling nanoscale circuits

A. Markov Random Field

Markov Random field (MRF) [2] 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. A markov random process has Markov Property 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 as a random variable χ_i has markov property iff it is conditionally dependent only on its neighbours.

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 neighbour of χ_i. Let us assume the Positivity condition, stated as in (1) and the condition for existence of Markovinity as given in (2).

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

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

Thus, as obvious from (1) and (2), MRF has positive mass or density and is pairwise Markov or we say, it posses local and global markov property. Now probability distribution of MRF can be written as in (3) and is known as **Gibbs Distribution**.

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

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, (3) can be derived from Hammersley-Clifford Theorem. This theorem can be stated as a probability distribution with positive mass or density, satisfies the local or global Markov Property w.r.t. 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).

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

Markov Random Field based probabilistic computing of nanodevices was proposed by R.I. Bahar [2]. 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 occurrence of logic zero and logic one. At the primary output, joint probability is calculated and for each intermediate node marginal probability is calculated using belief propagation algorithm [**?**], [17]. For a boolean network modeled as MRF, the clique energy is computed as summation over the minterms of the valid states. For example, for a two input XOR gate with input 01 and output 1 is called a Valid state (Vs), while input 01 and output 0 is called an invalid state (IVs). The energy of invalid states should be greater than the valid state's energy [2]. For a two input XOR gate with output x_2, the function $F(x_0, x_1, x_2) = 1$ when $x_2 = x_0\bar{x_1} + \bar{x_0}x_1$. The clique energy $E(\eta_i)$ for two input XOR gate, given below,can be computed using NANOLAB tool, with logic compatibility table of XOR gate given as input.

$$\begin{aligned} E(\eta_i) \quad = \quad & -2 + x_2 + x_1 - 2x_1x_2 + x_0 - 2x_0x_2 \\ & -2x_0x_1 + 4x_0x_1x_2 \end{aligned}$$

Bhaduri and Shukla [3], [8] designed a MATLAB based tool called NANOLAB based on MRF theory for probabilistic computation at nanoscale. The tool provides n-Modular Redundancy (NMR) approaches like TMR, CTMR, etc and NAND Multiplexing [**?**] to deal with unreliable nature of the device. Reliability of defect tolerant circuit was evaluated in presence of noise. The NANOLAB tool was evaluated for structural faults and bridge faults and results were compared with that of deterministic approach in [7].

Nepal and Bahar [**?**], [11]–[13] 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 [11]. The reliability is thus achieved on the cost of size. For example, a four transistor NAND Gate implementation requires 60 transistors by MRF approach suggested by Nepal [11]. Further in [12], [13] the modified clique energy function was used and new implementation of 2 input NAND Gate required only 28 transistors. However, as shown in Table 4 of [13] the power is greatly reduced (by 33%) for larger circuits by MRF mapping. Also, error correcting codes technique (Hamming code (6,3)) is introduced to enhance reliability at lower power with reduced device area as compared to previous works of Nepal *et. al* [11]. The feedback path reinforces the logic values and the paper claims to fix all one-bit deviations from correct codeword by reinforcing principle of MRF circuit. However with reconvergent fanouts, the circuit requires more runtime.

We have evaluated Markov Random Field based approach for combinational and sequential circuits in MATLAB. All 2-input logic gates are designed using this approach. A 8-bit full adder is implemented using Markov Random Field and the results are included in Section III.

B. Bayesian Networks

In place of Markov Random Field approach, which makes use of undirected graph another probabilistic approach based on Bayesian Networks (BNs) has been proposed by Bhanja et. al [4], [5], [10], [14]–[16]. Bayesian Networks are graphical probabilistic models representing the joint probability function over a set of random variables, using a directed acyclic graph structure (DAG). The nodes of this DAG represent random variables and node to node arcs denote direct conditional dependencies. Both Bayesian Network and Markov Random Field makes use of belief propagation algorithm to calculate joint and marginal probabilities for primary output and intermediate nodes. A conditional dependency is associated with nodes in both the cases. Markov Random Field based probabilistic model is undirected, whereas Bayesian Network is directed. Both Bayesian Network and Markov Random Field can generate Junction trees, but by using BN, we can arrive at smallest Junction Tree [15].

Each gate is modeled using conditional probability table (CPT). A CPT models probability of gate output signal being at a logic state, given its input signal states. Probabilities in a Bayesian Network model are computed using local message passing. The inference problem becomes complicated if the underlying undirected graph has cycles. The network compilation process ensures a loop free tree of cliques known as Junction Tree. The Junction Tree helps in local message passing on the cost of increased complexity. A Bayesian network is converted to moral graph and then it is triangulated. Finally a Junction Tree is obtained. The probabilistic modeling of nanoscale circuits as a Bayesian Network finds its application in computing output error probability and switching activity estimation. The Bayesian network preserves dependencies and can be applied to Networks having casual flow, for example Nano-Cmos, CNT, RTD, etc. The average case complexity of BN is least among all other probabilistic models. The worst case space complexity of BN is linear, i.e., $O(nF_{max})$ where n is number of nodes and f_{max} is maximum fan-in. The time complexity is also linear, i.e., $O(nN)$ where N is the number of samples for stochastic inference scheme. Time and space complexities of BN are independent of gate error probabilities. Hence we can say that bayesian network model is faster than other models.

C. Probabilistic Transfer Matrices

To estimate the effects of soft transient errors on logic circuits, another computational framework based on Probabilistic Transfer Matrices is proposed in literatures [6], [9]. This transfer matrix formulation represents parallel composition of logic gates with tensor products. Given individual gate error probabilities, we can compute output probabilities for the entire circuit and overall probability of correctness, i.e., reliability of circuit [9]. The Probabilistic Transfer Matrix and Ideal Transfer Matrix model the fault prone and fault free behavior of logic gates, respectively. A matrix representing all possible input probabilities can be computed by tensoring (Kronecker product) the input probabilities [6]. The process of combining gate Probabilistic Transfer Matrices implicitly takes into account signal dependencies between gates by considering the underlying joint and conditional probabilities within the circuit. Once the circuit Probabilistic Transfer Matrix is calculated, accurate information about output probabilities, reliability and signal observability can be extracted from it [6].

For a gate with n inputs and m outputs, PTM representation requires O(2n+m) space. For a computer with 2 GB memory, this limits the size of the circuit to 16 input/output signals only [6]. Such performance bottlenecks can be handled by using some matrix compression technique like Algebraic Decision Diagrams [9] or by applying some heuristic techniques like Dynamic Weighted Averaging Algorithm and Multi-pass approach [6] for approximate analysis. Still the memory usage is large or results are obtained at the cost of approximations. A SPR Tool [6] has been proposed that computes signal reliability of combinational circuit based on SPR model proposed in [6]. The current version of tool works with standard cell generated logic, where the intrinsic signal reliability of every cell is supposed to be known. The circuit description generated by the synthesis tool is analyzed for determining the cell logical ordering, reconvergent fanout signals and signals that will be considered in the reliability analysis. In brief, the SPR tool computes signal reliability of complete circuit and generates numerical results that can be used in a reliability driven design process or can be plotted in a numerical tool like MATLAB, for evaluation of reliability matrices like Mean Time Between Failure (MTBF), Failures In Time (FIT) or probability of correct operation [6].

We have designed the MATLAB code for Probabilistic Transfer Matrices and some combinational circuits are simulated. A 8-bit full adder is designed using Probabilistic Transfer Matrices approach. The results are provided in Section III

D. Probabilistic Decision Diagram

Another approach proposed by Afshin Abdollahi [1] makes use of Probabilistic Decision Diagrams for representing probabilistic behavior of circuits with faulty gates. Each gate has probabilistic behavior at nanoscale and this behavior is modeled as Probabilistic Decision Diagram which calculates the circuit reliability. As explained in [1], at output of each logic gate, a probabilistic inverter is attached to model the gates probabilistic behavior. Thus output function f changes to f' with probability p of probabilistic inverter. Abdollahi *et. al.* [1] describes how to construct Probability Decision Diagrams of different logic gates and circuits and extracting output probabilities and other information from PDDs. The Probability Decision Diagram is used to encode probabilities into the decision diagrams using weights of the edges. A Probabilistic Decision Diagram is a weighted graph with directed edges and has a single terminal node which represents constant $'0'$. The terminal node has no outgoing edges. All other nodes have two outgoing edges and a decision variable x. In Contrast to factor graph, each node and each edge of a Probabilistic Decision Diagram represents a function. The function f' of an edge is determined by $*$ operation on weight of edge p and function f of its end node. The $*$ operation on two probability values a and b is defined as:

$$a * b = (1 - a)b + a(1 - b)$$

The abdollahi et. al. [1] explains the construction of a Probabilistic Decision Diagram for a probabilistic circuit and extracting output error probabilities and other information from it.

We have implemented Probabilistic Decision Diagrams in MATLAB. The Probability Decision Diagrams for all logic gates are designed and simulated. These logic gate PDDs are used to make bigger circuits. The simulation results of a small circuit with two inputs and one output is provided in Section III.

III. Probabilistic Model Functional Evaluation

We have implemented Markov Random Field, Probabilistic Transfer Matrices and Probabilistic Decision Diagrams based methods using MATLAB. All 2-input logic gates are designed using these approaches. These logic gates are used for combinational circuit design. A 8-bit full adder is implemented using Markov Random Field and Probabilistic Transfer Matrices approaches. If we assume the probability of logic '1' and logic '0' on the primary input as 0.5 then, by using MRF based approach, the probability of obtaining correct sum and carry for a 1-bit adder is 0.4908. Here we assume no Gaussian noise is present. Again, if reliability of logic gates is assumed to be 0.9, then by using Probabilistic Transfer Matrices, the probability of sum and carry is given as:

$$\text{sum} = \begin{bmatrix} 0.41 & 0.09 \\ 0.09 & 0.41 \end{bmatrix} \text{carry} = \begin{bmatrix} 0.41 & 0.1525 \\ 0.4576 & 1.2299 \end{bmatrix}$$

A probabilistic transfer matrix P represents probability of occurrence of four states of a logic signal, i.e., $P_{00} = \text{correct0}$, $P_{01} = \text{incorrect0}$, $P_{10} = \text{incorrect1}$ and $P_{11} = \text{correct1}$. Table I gives the probability of getting logic '1' on Sum when inputs are $A = 11101010$ and $B = 11011111$ and $C_{in} = 0$. Here we assume that probability of getting correct logic value at primary inputs is 0.9.

All logic gates are implemented using Probabilistic Decision Diagram based approach in MATLAB. The example discussed in Fig. 9 of [1] is implemented and results are included in

TABLE I
NODE PROBABILITY OF SUM S OF 8-BIT ADDER USING MRF BASED APPROACH WHEN INPUTS ARE A=11101010 AND B=11011111 AND $C_in = 0$

	Probability of logic '1'	value
Sum(0)	0.724022	1
Sum(1)	0.172673	0
Sum(2)	0.282115	0
Sum(3)	0.686591	1
Sum(4)	0.260673	0
Sum(5)	0.284728	0
Sum(6)	0.682954	1
Sum(7)	0.739023	1

TABLE II
PROBABILITY DECISION DIAGRAM: RESULTS FOR EXAMPLE IN FIG. 9 OF [1]. HERE P = 0.1, Q = 0.2 AND R = 0.05

Node	F	X	Y
Left child probability	0.05	0.15	0
Right child probability	0.068	0.15	1

Table II. The edge weights of left and right child's of parent node F are taken as $p = 0.1$ and $q = 0.2$. The weight associated with parent node is $r = 0.05$.
Further, in Table III the Bayesian Network, Probabilistic Transfer Matrices and Probabilistic Decision Diagrams based approaches are compared for benchmark circuits. The data in this table is obtained from the referenced papers. It is observed that time complexity of Probability Decision Diagrams is better than Bayesian Network. The Probability Transfer Matrix based approach occupies alot of memory and takes long run time as compared to other approaches.

TABLE III
PERFORMANCE EVALUATION OF PROBABILISTIC APPROACHES

Circuits	BN time(s)	PTM time(s)	PDD time(s)
C17	0.0	0.076	0.001
pcle	0.07	74.9	0.002
count	1.14	-	0.010
alu4	1.87	-	1.049

IV. CONCLUSION

A brief overview of different probabilistic approaches for modeling nanoscale devices is presented here. The belief propagation algorithm used for message passing in some of these algorithms is also discussed. The probabilistic approaches like Markov Random Field, Probabilistic Transfer Matrices and Probabilistic Decision Diagrams are designed in MATLAB. Some combinational circuits are designed using these approaches. A 8-bit full adder is taken as a design example. The HUGIN tool is used for Bayesian networks and some basic circuits are simulated (the results are not included here). The Markov Random Field based approach can be extended to design sequential circuits. The n-modular redundancy can be easily added to MRF model. The CMOS realization of MRF approach is presented by Nepal *et.al.*. MRF approach uses undirected graphs and Bayesian Networks uses directed graph. It is observed that Bayesian Network and Probabilistic Decision Diagrams have least time complexity among these approaches. The Probabilistic Transfer Matrices require alot of memory and long simulation time.

REFERENCES

[1] A. Abdollahi, "Probabilistic decision diagrams for exact probabilistic analysis," in *Computer-Aided Design, 2007. ICCAD 2007. IEEE/ACM International Conference on*, 4-8 2007, pp. 266 –272.

[2] R. I. Bahar, J. Mundy, and J. Chen, "A probabilistic-based design methodology for nanoscale computation," in *In International Conference on Computer-Aided Design*, 2003, pp. 480–486.

[3] D. Bhaduri and S. Shukla, "Nanolab: A tool for evaluating reliability of defect-tolerant nano architectures," in *in IEEE Computer Society Annual Symposium on VLSI (IEEE*. Press, 2004, p. 0309.

[4] S. Bhanja and N. Ranganathan, "Switching activity estimation of vlsi circuits using bayesian networks," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 11, no. 4, pp. 558 – 567, aug. 2003.

[5] ——, "Cascaded bayesian inferencing for switching activity estimation with correlated inputs," *IEEE Trans. VLSI Syst.*, vol. 12, no. 12, pp. 1360–1370, 2004.

[6] D. T. Franco, M. C. Vasconcelos, L. Naviner, and J.-F. Naviner, "Signal probability for reliability evaluation of logic circuits," *Microelectronics reliability*, vol. 48, no. 8-9, pp. 1586 – 1591, 2008.

[7] M. S. Gaur, R. Narasimhan, V. Laxmi, and U. Kumar, "Structural fault modelling in nano devices," in *NanoNet*, 2008, pp. 6–10.

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

[9] S. Krishnaswamy, G. F. Viamontes, I. L. Markov, and J. P. Hayes, "Accurate reliability evaluation and enhancement via probabilistic transfer matrices," in *Design, Automation and Test in Europe Conference and Exposition (DATE 2005), 7-11 March 2005, Munich, Germany*. IEEE Computer Society, 2005, pp. 282–287.

[10] K. Lingasubramanian and S. Bhanja, "Probabilistic maximum error modeling for unreliable logic circuits," in *GLSVLSI '07: Proceedings of the 17th ACM Great Lakes symposium on VLSI*. New York, NY, USA: ACM, 2007, pp. 223–226.

[11] K. Nepal, R. I. Bahar, J. Mundy, W. R. Patterson, and A. Zaslavsky, "Designing logic circuits for probabilistic computation in the presence of noise," in *DAC '05: Proceedings of the 42nd annual Design Automation Conference*. New York, NY, USA: ACM, 2005, pp. 485–490.

[12] ——, "Designing mrf based error correcting circuits for memory elements," in *DATE '06: Proceedings of the conference on Design, automation and test in Europe*. 3001 Leuven, Belgium, Belgium: European Design and Automation Association, 2006, pp. 792–793.

[13] ——, "Designing nanoscale logic circuits based on markov random fields," *J. Electron. Test.*, vol. 23, no. 2-3, pp. 255–266, 2007.

[14] T. Rejimon and S. Bhanja, "Scalable probabilistic computing models using bayesian networks," in *Circuits and Systems, 2005. 48th Midwest Symposium on*, 7-10 2005, pp. 712 –715 Vol. 1.

[15] T. Rejimon, K. Lingasubramanian, and S. Bhanja, "Probabilistic error modeling for nano-domain logic circuits," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 17, no. 1, pp. 55 – 65, jan. 2009.

[16] T. Rejimon and S. Bhanja, "An accurate probalistic model for error detection," in *VLSID '05: Proceedings of the 18th International Conference on VLSI Design held jointly with 4th International Conference on Embedded Systems Design*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 717–722.

[17] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," in *Exploring artificial intelligence in the new millennium*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003, pp. 239–269.