Probabilistic Error Model for Unreliable Nano-logic Gates

Thara Rejimon and Sanjukta Bhanja
University of South Florida, Tampa, FL, USA.
E-mail: (rejimon, bhanja)@eng.usf.edu

Abstract—

We propose a novel formalism, based on probabilistic Bayesian networks, to capture, analyze, and model dynamic errors at nano logic for probabilistic reliability analysis. It will be important for circuit designers to be able to compare and rank designs based on the expected output error, which is a measure of reliability. We propose an error model to estimate this expected output error probability, given the probability of these errors in each device. We estimate the overall output error probability by comparing the outputs of an ideal logic model with a dynamic error-encoded model. We use of Bayesian inference schemes for propagation of probabilities. Since exact inference is worst case NP-hard, we use two approximate inference schemes based on importance sampling, namely EPIS(Evidence Pre-propagated Importance Sampling) and PLS (Probabilistic Logic Sampling), for handling mid-size benchmarks having up to 3500 gates. We demonstrate the efficiency and accuracy of these approximate inference schemes by comparing estimated results with logic simulation results.

I. Introduction

The ITRS road-map predicts CMOS device dimensions to reach close to the design limit of 50 nm by 2020. Circuits built with such nano-dimensional devices will face design challenges that have not been much of an issue so far. One such challenge involve dynamic errors in the interconnects and gates.

What is a dynamic error? These errors arise due to temporary malfunction of nano-devices while operated near thermal limits. These errors are significant in nano-computing due to very low noise margin, reduced supply voltages and low stored charges in nodes. We term these errors as dynamic errors since they are not permanent damages. *What complicates the picture is that this propensity for errors will, intrinsically, exist at each gate.* These errors are going to the measure of reliability in the nano devices. Hence, in future, reliable computation has to be achieved with "systemic" unreliable devices [1]. Thus making the entire computation process probabilistic rather than deterministic in nature. For instance, given inputs 1 and 0, an AND gate will output the state 0, only with probability $1-p$, where p is the error probability. Thus, traditional, deterministic, truth-table based logic representation will not suffice. Instead, the output needs to be specified in terms of probabilities, conditioned on the states of the inputs. There will be need for formalisms to compare, evaluate, and vet circuit designs with these dynamic error prone gates.

The **sources of dynamic errors**: Dynamic error will arise due to the use of ultra low voltage design, resulting in supply to ground bounce, leakage and coupling noise and due to small device geometry operating with only a handful of electrons even in next generation CMOS. Emerging devices like nano-wire FET, CNT-FET, resonant tunnel diodes(RTD) and would have increased effect of dynamic errors that will arise due to their operating conditions near thermal limit, due to background charge fluctuations and phonon coupling. Each of these devices would pose various design, modeling challenges for analyzing them: for example QCA is not really causal, CNT circuits have to operate without multiple fanouts.

The problem of reliable computing using unreliable devices is not new. Indeed the basic methods of Triple Modular Redundancy (TMR) and NAND Multiplexing were proposed in the 1950s [1]. There are works done on computation of error bounds [3], [1], [13] for noisy gates which are useful for higher level designs. At logic level, Han *et al.* [4] discussed various redundancy schemes for reliability enhancement. Patel *et al.* [2] provided a matrix-based formalism for modeling dynamic errors at logic level: scalability of such a technique is in question as the largest circuit handled had 46 gates. Order of magnitude improvement was reported by Rejimon *et al.* [14] using Bayesian formalism for error computation. Probabilistic model checking for reliability-redundancy trade-off was inroduced in [6]. Markov models are proposed in [5] for signal probabilities for circuits that are extremely small and did not have any re-convergence. We know that error probability inference is NP-hard, however, using conditional independence and smart approximation, reasonable estimates can be found. In this work, we present an approximate algorithm for the error probability computation for mid-sized ISCAS benchmarks.

Suppose, we have logic block with inputs $Z_1, \cdots Z_N$, internal signals $X_1, \cdots X_M$, and outputs $Y_1, \cdots Y_K$. Let the corresponding versions of the internal lines and the outputs with dynamic errors be denoted by $X_1^e, \cdots X_M^e$ and $Y_1^e, \cdots Y_K^e$, respectively. Thus, the error at the ith output can be mathematically represented as the XOR of the error-free and the output with error.

$$E_i = Y_i^e \oplus Y_i \tag{1}$$

We propose the output error probability $P(E_i = 1) = P(Y_i^e \bigoplus Y_i = 1)$ as a design metric, in addition to other traditional ones such as area or delay, to vet

different designs in the nano-domain. Note that the probability of output error is dependent on the individual gate error probability p and also on the internal dependencies among the signals that might enhance or reduce the overall output error based on the circuit structure. We model these dependencies by a **Bayesian Network**, which is known to be the exact, minimal probabilistic model for the underlying joint probability density function (pdf) for *causal* logic networks. Probabilistic belief propagation on these Bayesian Networks can then to be used to estimate this probability.

In this work, we report two approximate inference schemes (1) Probabilistic Logic Sampling and (2) Evidence pre-propagated importance sampling for the approximate inference for the mid-size benchmarks. In the absence of nano-domain benchmarks, we use mid-size ISCAS benchmarks having gate count up to 3500 to show the scalability, accuracy and efficiency of our modeling using these approximate inference schemes.

II. Bayesian Networks

Bayesian Networks are graphical probabilistic models representing the joint probability function over a set of random variables using a directed acyclic graphical structure (DAG), whose nodes describe random variables and node to node arcs denote direct causal dependencies. In a Bayesian network, the exact joint probability distribution over a set of n variables, $X_1 \cdots, X_n$ in this network is given by Eq. 2.

$$P(x_n, x_{n-1}, \cdots, x_2, x_1) = P(x_n|x_{n-1}, \cdots, x_1) \\ P(x_{n-1}|x_{n-2}, \cdots, x_1) \cdots\cdots P(x_1) \quad (2)$$

Any random variable, X_k is independent of all other variables, given the states of its parent nodes, say, X_{k-1} and X_{k-2}. This *conditional independence* can be expressed by Eq. 3.

$$P(x_k|x_n, x_{n-1}, \cdots, x_1) \quad = \quad P(x_k|x_{k-1}, x_{k-2})) \quad (3)$$

Mathematically, this is denoted as $I(X_k, \{X_{k-1}, X_{k-2}\}, \{X_n, \cdots, X_{n-1}\})$. Using the conditional independence in directional graph, we arrive at an optimal factorized form that involve conditional probabilities based on the parents (or direct causes, inputs) to a node (effect, output): $P(X) = \prod_{k=1}^{m} P(x_k|pa(x_k))$. Even though probabilistic inference is worst-case NP-Hard, these factorized forms can reduce complexity significantly for general cases.

III. Probabilistic Error Model

We compute the error probability $P(e_i) = P(Y_i^e \bigoplus Y_i = 1)$ by marginalizing the joint probability function over the inputs, internal lines, and the outputs

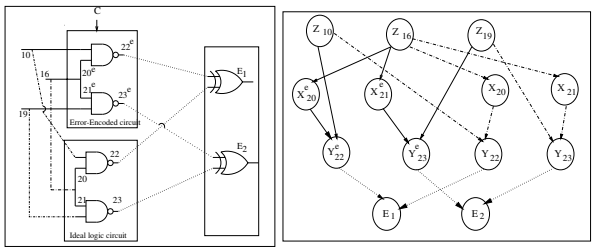

Fig. 1. (a) Conceptual circuit representation of the logic used to detect errors involving the ideal logic and the unreliable logic components. (b) The corresponding Bayesian network representation.

$$\begin{aligned} P(e_i) &= \sum_{z_1, \cdots z_N} P(e_i|z_1, \cdots z_N) P(z_1) \cdots P(z_N) \\ &= P(z_1) \cdots P(z_N) \sum_{z} \sum_{x, x^e} P(e_i|z_1, \cdot z_N) \\ &= P(z_1) \cdots P(z_N) \\ &\quad \sum_{\forall z} \sum_{\forall x, \forall x^e} P(e_i, y_i, y_i^e, z_1, \cdot z_N, x_1, \cdots x_M, x_1^e, \cdots x_M^e) \end{aligned}$$

where Eq. 4 shows that the joint density function that is necessary to compute the dynamic error exactly. Summing over all possible values of all the involved variables is computationally expensive (NP-hard), hence we require a graphical model that would use the causal structure and conditional independences to arrive at the minimal optimally factorized representation of this joint probability function as a Bayesian network.

A. The Bayesian Network Structure

We model, both error-free logic and the logic with dynamic errors, as a Directed Acyclic Graph (DAG). These two models, which we will refer to as the ideal logic model and the error-encoded model, are then connected at the outputs by comparators. The comparator output is the variable $E_i = Y_i^e \oplus Y_i$ in Eq. 1. The comparator output of logic 1 indicates that the ideal logic model and error-encoded model outputs are different. The probability of the comparator outputs being in state *"1"* provides the overall output error probability, $P(E_i = 1)$ of output Y_i of the circuit.

Figure 1(a) shows the (conceptual) representation of a error detection circuit for a simple logic involving two NAND gates, represented by block C. The other block involves the same logic, but built with unreliable components. These gates are assumed to have gate error probability of p. The inputs to both the blocks are the same. The two outputs are connected to two comparators. The output of the comparator represents error in computation. Note that this is just a conceptual representation, we do not actually propose synthesizing the circuit. From the conceptual

circuit design, we can construct the Bayesian network representation, which we call the LIPEM-DAG model. Each node in the LIPEM is a line in circuit and the links denote a connection between the lines via gates. Figure 1(b) shows the LIPEM corresponding to the circuit in Figure 1(a).

B. Computing the Error Probability

We used a stochastic algorithm, based on importance sampling, to compute the error probability based on the built LIPEM model. This approximate scheme can be proven to converge to the correct probability estimates [10]. The stochastic importance sampling algorithm generates randomly selected instantiations of the network variables in topological order, according to probabilities in the model, and then calculates frequencies of instantiations of interest to obtain estimates of the probabilities. Given the states of certain nodes in the network (evidence) and the conditional probability table of each node, this algorithm generates sample instantiations of the network so as to generate state of each node in the network. From these sample instantiations, it can find approximate probabilities of each state for each node in the network. This sampling is done according an the optimum importance function that closely resembles the underlying joint probability distribution. **Probabilistic Logic Sampling (PLS):** Probabilistic logic sampling is the first and the simplest sampling algorithm proposed for Bayesian Networks [10]. The salient features of these algorithms are: (1) they scale extremely well for larger systems making them a target inference for nano-domain billion transistor scenario and (2) they are any-time algorithm, providing adequate accuracy-time trade-off and (3) The samples are not based on inputs and the approach is input pattern insensitive. The flow of the algorithm is as follows.

1. Complete set of samples are generated for the Bayesian network using the optimal importance function, which is the joint probability distribution function $P(X)$. The importance function is never updated once it is initialized. This assumption makes sense when the child node evidences are not present.
2. In case of child node evidences (important in diagnostic backtracking), samples that are incompatible with the evidence set are disregarded.
3. The probability of all the query nodes are estimated based on counting the frequency with which the relevant events occur in the sample.

Evidence Pre-propagated Importance Sampling (EPIS): The evidence pre-propagated importance sampling (EPIS) [7], [8] uses local message passing and stochastic sampling. This method scales well with circuit size and is proven to converge to correct estimates. This is also an anytime-algorithm since it can be stopped at any point of time to produce estimates. Of course, the accuracy of estimates increases with time. Like PLS, EPIS is also based on importance sampling that generates sample instantiations of the *whole* DAG network, i.e. for all line states in our case. These samples are then used to form the final estimates. The difference is with respect to the importance function used for sampling. EPIS takes into account any available evidence. In a Bayesian network, the product of the conditional probability functions at all nodes form the optimal importance function. Let $X = \{X_1,\ X_2, \cdots, X_m\}$ be the set of variables in a Bayesian network, $Pa(X_k)$ be the parents of X_k, and E be the evidence set. The importance function can be approximated as

$$P(X|E) = \prod_{k=1}^{m} \alpha(Pa(X_k))P(x_k|Pa(X_k))\lambda(X_k) \quad (4)$$

where $\alpha(Pa(X_k)) = (P(E^-|Pa(X_k)))^{-1}$ is a normalizing constant dependent on $Pa(X_k)$ and $\lambda(X_k) = P(E^-|x_k)$, with E^+ and E^- being the evidence from parents and children, respectively, as defined by the directed link structure. Calculation of λ is computationally expensive and for this, Loopy Belief Propagation (LBP) [9] over the Markov blanket of the node is used. Yuan *et al.* [8] proved that for a poly-tree, the local loopy belief propagation is optimal. The importance function can be further approximated by replacing small probabilities with a specific cutoff value [7].

IV. Experimental Results

The approximate computation of the LIPEM using Bayesian Networks was done by a tool named "GeNIe" [11]. We used PLS and EPIS. The tests were performed on a Pentium IV, 2.00GHz, Windows XP computer. We show that the error model and associated computations scales extremely well with circuit size by showing results with circuits of varying sizes. Table I shows the average output error probabilities for various ISCAS'85 benchmark circuits for three different gate errors of $p = 0.01, 0.001$ and 0.0001. Reported results were obtained from PLS and EPIS with 1000 samples. As expected, all circuits exhibit higher overall error as individual gate error increases. It can be observed that c499 has lower error growth over all the other circuits. For many of the benchmark circuits (namely c432, c1908, c3540, c6288, c7552), the average output error exceeds 0.1 for gate error probability 0.01. This indicates that 0.01 is unacceptable gate error probability for these circuits. Additional gate level redundancy may be required for these circuits.

To validate our model we performed an in-house logic simulation of the benchmarks with 500,000 random vectors obtained by changing seed after every 50,000

TABLE I

OUTPUT ERROR PROBABILITIES FOR ISCAS'85 CIRCUITS.

	No. of gates	Average Output error Probability for individual gate error probability p					
		=0.01		=0.001		=0.0001	
		PLS	EPIS	PLS	EPIS	PLS	EPIS
c432	160	0.119	0.118	0.016	0.017	0.0021	0.0008
c499	202	0.030	0.031	0.004	0.004	0.0005	0.0002
c880	383	0.078	0.074	0.009	0.008	0.0013	0.0010
c1355	546	0.059	0.060	0.007	0.007	0.0009	0.0009
c1908	880	0.134	0.130	0.024	0.023	0.0080	0.0071
c2670	1193	0.075	0.077	0.018	0.018	0.0087	0.0100
c3540	1669	0.257	0.258	0.127	0.121	0.0965	0.0918
c5315	2307	0.075	0.076	0.010	0.009	0.0017	0.0010
c6288	2416	0.386	0.388	0.111	0.105	0.0181	0.0150
c7552	3512	0.111	0.109	0.018	0.018	0.0024	0.0021

TABLE II

COMPARISON OF BAYESIAN NETWORK MODELING AND LOGIC SIMULATION

	BN-Nodes	PLS		EPIS		Sim. Time
		μ_e	T1(s)	μ_e	T2(s)	T_{sim}(s)
c432	475	0.0020	0.234	0.0030	0.561	19.763
c499	565	0.0016	0.344	0.0016	0.912	23.013
c880	956	0.0023	0.844	0.0019	1.773	43.147
c1355	1253	0.0020	1.141	0.0019	3.245	58.999
c1908	2172	0.0081	2.187	0.0076	9.503	95.030
c2670	3097	0.0095	3.205	0.0094	55.30	144.300
c3540	4038	0.0807	4.547	0.0759	82.60	192.060
c5315	6247	0.0073	7.844	0.0023	155.55	295.162
c6288	4896	0.0058	5.672	0.0073	118.18	297.473
c7552	8398	0.0236	11.64	0.0274	233.08	445.326

vectors and obtained the average output error probabilities for different gate error probabilities. In Table II, we compare simulation results with Bayesian network results. We report the accuracy of our model in terms of average error, μ_e , between the exact output error probabilities and the estimated output error probabilities obtained from PLS and EPIS with 1000 samples. Mean estimation error (over all circuits) for PLS is 0.012 and for EPIS is 0.011. Average estimation time taken by PLS for 1000 samples is found to be 3.72 sec. whereas average time taken by EPIS for the same number of samples is 31 sec. and the average simulation time is 161.43 seconds. These results show the effectiveness of our model in terms of accuracy and estimation time. It is observed from the results that accuracy of forward inference with PLS is almost the same as that of EPIS, but PLS has less time complexity. However for backward reasoning problems like input space characterization, EPIS is the only choice.

Comparing Designs: Figure 2 shows the variation in average output error with gate error p for two ISCAS benchmark c499 and c1355 that are logically equivalent. We see that c499 is clearly a better design of the logic for nano-domain in terms of resistance to dynamic errors. The circuit c1355 is more sensitive to dynamic-error almost for all individual gate error probabilities. For example, when p=0.01, the average output error probability become 0.06 as opposed to 0.03 for c499. The expected output dynamic error can be used, along with other design measures such as power and area for nano-domain circuits.

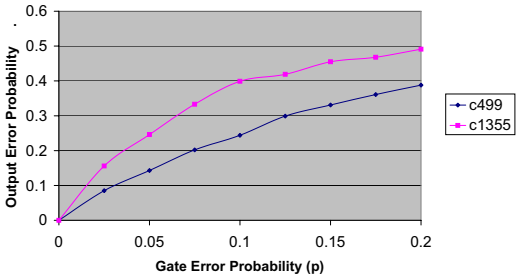

Fig. 2. Output error profiles of two alternative logic implementation (c499, and c1355).

In conclusion, we presented an exact probability model, based on Bayesian networks, to capture the inter-dependent effects of dynamic errors at each gate. Dynamic error at each gate is modeled through the conditional probability specifications in the Bayesian network. The expected output error, also the measure of realiability, can be used to vet design choices. We used two scalable, any-time approximate inference schemes to compute probability of output error. Comparing estimated results with simulation results, we show that the estimation is close-to-exact. We demonstrate scalability of our estimation tool by using the mid-sized ISCAS'85 CMOS benchmarks (3000 gates), which would be equally scalable for other nano-devices. We are currently working on modeling dynamic error tolerant designs by applying TMR redundancy on selected nodes having high dynamic error sensitivities, and also on selection of input space for a desired output behavior (Bayesian backtracking).

REFERENCES

[1] J. von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," in *Automata Studies* (C. E. Shannon and J. McCarthy, eds.), pp. 43–98, Princeton Univ. Press, Princeton, N.J., 1954.

[2] K. N. Patel, I. L. Markov, and J. P. Hayes, "Evaluating Circuit Reliability Under Probabilistic Gate-Level Fault Models", *International Workshop on Logic and Synthesis*, May 2003.

[3] J. B. Gao, Yan Qi and J.A.B. Fortes,"Bifurcations and Fundamental Error Bounds for Fault-Tolerant Computations" *IEEE Transactions on Nanotechnology*, vol. 4-4 pp. 395–402, July 2005.

[4] J. Han, J. B. Gao,P. Jonker, Yan Qi and J.A.B. Fortes, "Toward hardware-Redundant Fault-Tolerant Logic for Nanoelectronics" *IEEE Transactions on Design and Test of Computers*, vol. 22-4 pp. 328–339, July-Aug. 2005.

[5] J. Chen *et al.*, "A Probabilistic Approach to Nano-computing," *Workshop on Non-Silicon Computation*, June 2003.

[6] G. Noman, *el al.*, "Evaluating the reliability of defect-tolerant architectures for nanotechnology with probabilistic model checking", *International Conference on VLSI Design*, 2004.

[7] J. Cheng, "Efficient Stochastic Sampling Algorithms for Bayesian Networks," *PhD Dissertation, University of Pittsburgh*, 2001.

[8] C. Yuan and M. J. Druzdzel, "An Importance Sampling Algorithm Based on Evidence Pre-propagation," *Uncertainty on Artificial Intelligence*, pp. 624-631, 2003.

[9] K.P. Murphy, Y. Weiss and M.I. Jordan "Loopy belief propagation for approximate inference: an empirical study," *In Proceedings of Uncertainty in AI*, pp. 467–475, 1999.

[10] M. Henrion, "Propagation of uncertainty by probabilistic logic sampling in Bayes' networks," *Uncertainty in AI*, 1988.

[11] ''GeNIE'' URL http://www.sis.pitt.edu/~genie/

[12] J. Pearl, "Probabilistic Reasoning in Intelligent Systems: Network of Plausible Inference", Morgan Kaufmann Publishers, Inc., 1988.

[13] N. Pippenger, "Reliable Computation by Formulas in the Presence of Noise", *IEEE Trans on Inf. Theory*, vol. 34(2), pp. 194-197, 1988.

[14] T. Rejimon and S. Bhanja, "Scalable Probabilistic Computing Models using Bayesian Networks", *IEEE Midwest Symposium on Circuits and Systems*, pp. 712-715, July 2005.