A Framework for Reliability Analysis of Combinational Circuits Using Approximate Bayesian Inference

Shivani Bathla, *Graduate Student Member, IEEE*, and Vinita Vasudevan, *Member, IEEE*

***Abstract*— A commonly used approach to compute the error rate at the primary outputs (POs) of a circuit is to compare the fault-free and faulty copies of the circuit using XOR gates. This model results in poor accuracies with nonsampling-based methods for reliability estimation. An alternative is to use a single copy of the circuit with a four-valued representation for each net corresponding to the correct and incorrect signals. One problem in this formulation is the accurate propagation of associated probabilities. We use the framework of Bayesian inference (BI) to address this issue. We derive the conditional probability distribution (CPD) corresponding to the four-valued signals and find the output error rate using various approximate BI techniques. With our formulation, we demonstrate that the output error rate scales with the gate error probabilities. It is guaranteed to be zero when the gate error probability is zero, provided approximate BI algorithms based on sum-product belief propagation (BP) are used. Although inaccuracies increase at very low gate error probabilities, it is able to capture the relative reliability of outputs with respect to each other. We also propose a new method for finding the overall circuit error rate as the partition function for a fixed state of POs. This method provides a significant improvement in accuracy when compared with the existing method using OR gates.**

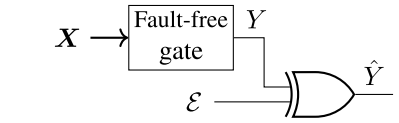
***Index Terms*— Bayesian inference (BI), Bayesian networks (BNs), error rate, reliability, signal probability.**

I. Introduction

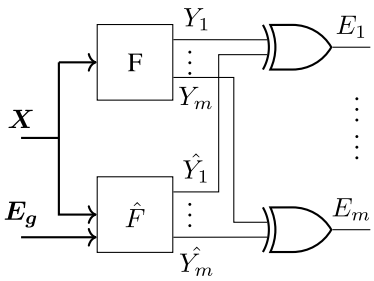
AGGRESSIVE scaling of CMOS technology has led to a sharp increase in manufacturing defects and transient faults in gates due to low threshold voltages, process variations, electromigration, and crosstalk. This has resulted in unreliable logic gates, whose outputs are not completely determined by the inputs to the gate. Rather, given the inputs, one can only specify the probability that the output of a gate is a zero or a one. This probabilistic behavior can also occur due to the aging of circuits. It is exploited by imprecise circuits for error-resilient applications to get energy savings.

The most commonly used model for an unreliable gate is the Von Neumann error model [1]. It is essentially a model of a binary symmetric channel in which the probability of getting a one instead of zero is the same as that of getting a zero instead of one. The errors in each gate are assumed to be independent of each other. At the gate level, this model is equivalent to representing a faulty gate as a fault-free gate followed by an XOR gate, with the other input of the XOR gate connected to an independent source of error ($\mathcal{E}$), as shown in Fig. 1(a).

The outputs of circuits designed using these unreliable gates have a nonzero probability of error. The design, synthesis, and optimization of these circuits require efficient methods to compute this error probability. This computation involves a comparison of the outputs of the error-free (F) and faulty

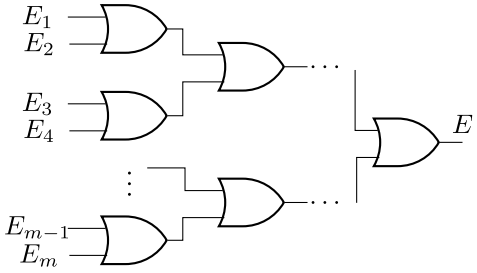


(a) Model of a faulty gate.



$\boldsymbol{E_g} = \{\mathcal{E}_1, \ldots, \mathcal{E}_N\},\ \boldsymbol{E} = \{E_1, \ldots, E_m\}$

(b) Circuit model used to compute error rate at outputs



(c) Miter used to compute circuit error rate

Fig. 1. Models for reliability analysis of circuits. F denotes the error-free circuit, and $\hat{F}$ denotes the faulty circuit. $\mathcal{E}_i$ denotes the independent source of error for the ith gate in the faulty circuit, and $\boldsymbol{E_g}$ is the set of all error sources. E_i denotes the error signal corresponding to the ith PO, and E denotes error in at least one of the POs. (a) Model of a faulty gate. (b) Circuit model used to compute error rate at outputs. (c) Miter used to compute circuit error rate.

Manuscript received 12 December 2022; revised 9 January 2023; accepted 14 January 2023. Date of publication 26 January 2023; date of current version 22 March 2023. *(Corresponding author: Shivani Bathla.)*

The authors are with the Department of Electrical Engineering, IIT Madras, Chennai 600036, India (e-mail: ee13s064@ee.iitm.ac.in; vinita@ee.iitm.ac.in).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TVLSI.2023.3237885.

Digital Object Identifier 10.1109/TVLSI.2023.3237885

circuit ($\hat{F}$) using XOR gates, as shown in Fig. 1(b). The most commonly used metric for evaluation of these circuits is the *error rate* at each primary output (PO) of the circuit [2], [3], [4], [5]. It is defined as the probability of error at a particular output, averaged over all possible values of the primary inputs (PIs). It is computed as the signal probability of the outputs of the XOR gates $\{E_1, E_2, \ldots, E_m\}$ in Fig. 1(b), with the signal probabilities of the PIs ($\boldsymbol{X}$) set to 0.5. The other metric that is used is the *circuit error rate*, which is the probability of getting an error in at least one of the outputs. This is typically computed after connecting a tree of OR gates to the XOR gates, as shown in Fig. 1(c).

The problem of reliability analysis is therefore a problem of computing signal probabilities in a system that has both fault-free and faulty circuits. The computation of signal probabilities in a circuit is known to be #P complete [6] and is difficult even in error-free circuits. Reliability analysis is significantly more difficult due to the additional signals and gates needed to model erroneous circuits, as well as due to the additional reconvergent loop created by connecting XOR gates at the output. Exact methods for signal probability computation include methods based on probabilistic transfer matrices (PTMs) [7], [8], probabilistic gate models (PGMs) [9], binary decision diagrams (BDDs) [3], [10], weighted model counting (WMC) [3], and Bayesian networks (BNs) [11], [12]. However, the associated exponential time/space complexity limits the applicability of these methods to relatively small circuits.

Several approximate methods have been proposed in the literature. The simulation-based methods include logic simulation using the Monte Carlo (MC) framework, stochastic computation model (SCM) [13], and sampling-based Bayesian inference (BI) methods [5], [14]. In these methods, each erroneous gate requires generation of an additional random number stream. Since the accuracy depends on the number of samples used, the time complexity increases as the gate error probabilities reduce. Therefore, at low gate error probabilities, the sampling-based techniques become unsuitable for use within an optimization framework that requires error rate computations in each iteration. Another disadvantage is that these methods are inflexible in the sense that any change to the circuit requires a complete reevaluation.

An alternative is to use deterministic approximate methods based on signal probability computation. The methods proposed in [9] and [15] assume that the inputs to a gate are independent and thus have limited accuracy. To improve accuracy, signal correlation coefficients proposed in [16] and [17] are used in the methods proposed in [4], [18], [19], and [20].

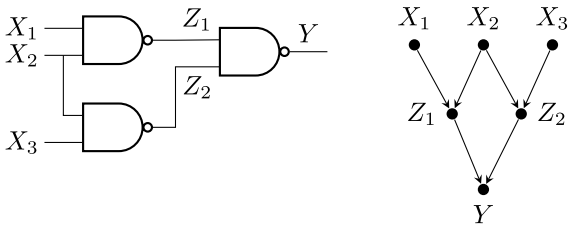
The main challenge in deterministic approximate techniques is accounting for correlations between nets. This includes the correlations between the error-free and corresponding erroneous nets as well as the correlations due to reconvergent fanouts. The approximate analysis methods proposed in [4], [15], [21], [22], [23], [24], [25], and [20] have an interesting feature. Instead of using two copies of the circuit, they use a single copy in which each net is associated with additional probabilities that take into account the error-free and erroneous values of the signal. These methods are attractive since they not only remove the additional reconvergent loop created by the XOR gates but also obviate the requirement for maintaining correlations between the error-free and corresponding erroneous nets. In [4], every net is associated with two conditional probabilities that give the probability of occurrence of error in a net, given its correct value. In the methods proposed in [15], [20], [21], [22], [23], [24], and [25], each net is modeled as a four-valued random variable, corresponding to its erroneous and error-free values. The associated probabilities are the joint probabilities of the error-free and the corresponding erroneous net. The problem here is the propagation of the four probabilities. The existing methods either assume independence between gate inputs [15], [21] or compute signal correlation coefficients [4], [20] or reliability correlation coefficients [23], [24], [25]. The main issue with these methods is thus the inaccuracies that arise due to the estimation and propagation of these correlations. Typically, pairwise signal correlation coefficients are used, further limiting the accuracy.

In this article, we propose a novel algorithm for reliability analysis based on approximate BI. As in [15], we use a single copy of the circuit, with each net modeled as a four-valued random variable with the associated probabilities. However, instead of propagating correlation coefficients, we cast the problem as a BI problem. For each gate in the circuit, we derive the conditional probability distribution (CPD) that determines the probability of each of the four states of the output, given all possible states of the inputs. Approximate (and wherever possible, exact) inference methods are then used to find the error rate at the POs. Although sampling-based methods can also be used, the focus in this article is the approximate deterministic methods.

With our formulation, we demonstrate that the output error rate scales with the gate error probabilities. It is also guaranteed to be zero when the gate error probability is zero, provided approximate BI algorithms based on sum-product belief propagation (BP) are used. This property does not hold good if the model in Fig. 1(b) is used for reliability computation. In this case, very often the computed error rate at the output is 0.5, which is just noise. In contrast, although inaccuracies do increase at low gate error probabilities, our method is able to capture the relative reliability of the outputs with respect to each other for gate error probabilities as low as 10^{-6}. Hence, it can be used to identify POs that are more susceptible to error.

We also propose a new formulation for finding the circuit error rate as the partition function corresponding to a fixed state of POs. This method gets rid of the additional OR gates connected to the XOR outputs in Fig. 1(c) and results in a significant improvement in accuracy. It also has very good run times and is suitable for use within an optimization routine.

The rest of this article is organized as follows. Section II has the notation used and the background on BI. We discuss the proposed formulation in Section III, the results obtained in Section IV, and a more detailed comparison with related work in Section V. Finally, we present our conclusions.



(a) (b)

$P(Z_1\|X_1X_2)$	$Z_1=0$	$Z_1=1$
$X_1X_2=00$	0	1
01	0	1
10	0	1
11	1	0

(c)

Fig. 2. (a) Sample circuit (b) BN model of the circuit and (c) CPD of an error-free NAND gate.

II. Background

A. Notation

We use the following notation. We use capital letters to denote random variables, small letters to denote the states or values that a random variable can take, and boldface letters to denote a set of random variables. Therefore, $P(Z = z)$ denotes the probability that the random variable Z takes on the value z. $\mathbf{X}$ and $\hat{\mathbf{X}}$ are used to denote the set of error-free and erroneous PIs. Similarly, $\mathbf{Y}$ and $\hat{\mathbf{Y}}$ denote the set of error-free and erroneous POs of the circuit, respectively.

B. Bayesian Networks

A BN is a probabilistic graphical model that captures the joint probability distribution (JPD) over a set of variables ($\mathbf{Z}$) using a directed acyclic graph (DAG). Each node in the DAG is a random variable, and edges capture the causal relationship between variables. The BN corresponding to a circuit is a DAG that has directed edges from each input to the output of a gate. The variables in the BN thus correspond to nets in the circuit that are either PI or outputs of gates in the circuit. Variables that are connected by an incoming edge to a variable Z_i are referred to as the *parents* of Z_i and denoted as Pa_{Z_i}. Each variable is associated with a CPD that specifies the probability of the variable, given the state of its parents. In circuit parlance, the CPD specifies the probability that the output of a gate takes on a particular value, given the values of the input. Fig. 2 shows an example of a BN for a circuit and the CPD for an error-free NAND gate.

A fundamental property satisfied by a BN is the following.

Property 1: Given the state of its parent variables, a variable in the BN is conditionally independent of all the nonsuccessors.

For the network in Fig. 2(b), this property would mean, for example, $P(Y|Z_1, Z_2, X_1) = P(Y|Z_1, Z_2)$ and $P(Z_1, Z_2|X_1, X_2) = P(Z_1|X_1, X_2)P(Z_2|X_1, X_2)$. As a result of this property, the JPD of all the variables in the network ($\mathbf{Z}$) can be written in a factorized form as follows:

$$P(\mathbf{Z}) = \prod_{Z_i \in \mathbf{Z}} P(Z_i \mid Pa_{Z_i}) \tag{1}$$

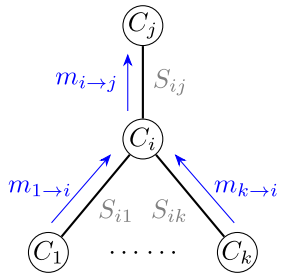


Fig. 3. Depiction of message-passing from cluster C_i to cluster C_j.

where Pa_{Z_i} are the parents of Z_i, $P(Z_i|Pa_{Z_i})$ is the CPD associated with Z_i, and $\mathbf{Z}$ is the set of all the variables in the BN. Each CPD is said to be a *factor* over the set of variables $\{Z_i \cup Pa_{Z_i}\}$, and the product of CPDs is computed as a *factor product* (defined in the Appendix). The probability distribution of a variable Z_i, $P(Z_i)$, is referred to as the marginal probability of Z_i. The *signal probability* of Z_i is defined as $P(Z_i = 1)$.

The task of computing the marginal probabilities of all the nets in the circuit is one of the inference tasks possible in a BN. The deterministic methods for approximate BI are variational techniques that optimize an energy functional using BP algorithms. In a majority of these methods, BP is performed using the sum-product BP algorithm [26, Ch. 11], which performs message-passing on an undirected *cluster graph*. Each node in this graph is associated with a *cluster* or a set of variables in the BN, and it is denoted as C_i. The edge between two clusters C_i and C_j is associated with a subset of variables contained in both the clusters, i.e., each edge is associated with a weight $S_{i,j} \subseteq C_i \cap C_j$. The weights $S_{i,j}$ are referred to as *sep-sets*. Each CPD of the BN, called the initial factors, is assigned to a single cluster in the undirected cluster graph that contains the corresponding variable and its parents. The BP algorithm is a message-passing algorithm in which a cluster C_i transmits a message to C_j based on messages from all its other neighbors. This is depicted in Fig. 3. The message $m_{i\to j}$ from cluster C_i to cluster C_j is computed using the following *sum-product* operation

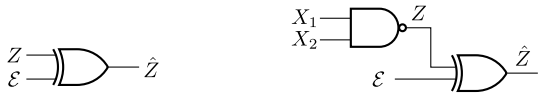
$$m_{i\to j}(S_{ij}) = \sum_{C_i \setminus S_{ij}} \psi_i \prod_{k \in \text{Neighbors}(i)\setminus j} m_{k\to i}(S_{ik}) \tag{2}$$

where ψ_i is the product of initial factors assigned to cluster C_i. The factor product of ψ_i and incoming messages from the neighbors of C_i is marginalized over variables that are not present in the sep-set $S_{i,j}$.

In exact inference techniques, the cluster graph is constrained to be a tree with sep-set variables $S_{i,j} = C_i \cap C_j$. This tree is called the *junction tree* or the *join tree* (JT). Exact inference requires two rounds of message-passing—a downward pass from a randomly chosen root node to the leaves of the tree and an upward pass back from the leaves to the root. The time and space complexity of exact inference using BP is exponential in the maximum cluster size. Circuits with long and nested reconvergent loops tend to have larger clusters, making exact inference infeasible. The approximate methods limit the maximum cluster size, but allow for cycles

$P(\hat{Z}\|Z)$	$\hat{Z}=0$	$\hat{Z}=1$
$Z=0$	$1-p_\epsilon$	p_ϵ
1	p_ϵ	$1-p_\epsilon$

$P(\hat{Z}\|X_1X_2)$	$\hat{Z}=0$	$\hat{Z}=1$
$X_1X_2=00$	p_ϵ	$1-p_\epsilon$
01	p_ϵ	$1-p_\epsilon$
10	p_ϵ	$1-p_\epsilon$
11	$1-p_\epsilon$	p_ϵ



(a) (b)

Fig. 4. Error model and the CPD for an erroneous signal and a faulty NAND gate. (a) Erroneous signal. (b) Faulty NAND gate.

in the cluster graph. These include loopy BP (LBP) [27] and its variants [28], [29], generalized BP (GBP) [30], and iterative join graph propagation (IJGP) [31]. In these methods, the message-passing algorithm is run iteratively on a loopy graph until convergence. The following properties are satisfied after convergence.

1) Each cluster is associated with a valid but possibly approximate JPD ($Q_i(C_i)$) which is computed as follows:

$$Q_i(C_i) = \psi_i \cdot \prod_{j \in \text{Neighbors}(i)} m_{j \to i}(S_{ij}). \tag{3}$$

2) The marginal probability of a variable is the same in all the clusters in which it is present.

Note that not all the BP algorithms use the exact sum-product operation to compute messages. One example is the weighted mini-bucket (WMB) method proposed in [32]. Another algorithm for approximate inference is the incremental build-infer-approximate technique (IBIA) [33] which uses a sequence of junction trees. The maximum cluster size in each of the junction-trees is set to a user-specified value. Sum-product BP is used in each junction tree to infer probabilities. In most of these algorithms, it is possible to trade off runtime for increased accuracy by having a larger number of variables in a cluster.

The other techniques used for inference are methods based on WMC [34]. These are meant for exact inference and can be used only for small circuits. There are also several sampling-based techniques such as probabilistic logic sampling (PLS) [14], evidence prepropagated importance sampling [35], and Gibbs sampling [36]. The accuracy obtained with these methods relies on the method used for sample generation and the number of samples used.

III. Proposed Formulation

A. Motivation

As discussed in Section I, the error model that is used for reliability analysis is the Von Neumann error model. If Z and $\hat{Z}$ denote the correct and incorrect values of the output of a gate and p_ϵ denotes the gate error probability, the CPD $P(\hat{Z}|Z)$ specified by this model is as shown in Fig. 4(a). The figure also has the corresponding gate-level representation of an erroneous signal, which is an XOR gate that has an independent error signal $\mathcal{E}$. p_ϵ is thus the signal probability of $\mathcal{E}$, i.e., $p_\epsilon = P(\mathcal{E}=1)$.

Based on this, the gate error model consisting of an error-free gate followed by an XOR gate [shown in Fig. 1(a)] can equivalently be represented as a CPD of the outputs, given the input values, denoted as $P(\hat{Z}|\boldsymbol{X})$. This can be derived as follows:

$$\begin{aligned} P(\hat{Z}|\boldsymbol{X}) &= \sum_{z,\epsilon} P(Z|\boldsymbol{X}) P(\hat{Z}|Z,\mathcal{E}) P(\mathcal{E}) \\ &= \sum_{z} P(Z|\boldsymbol{X}) P(\hat{Z}|Z). \end{aligned} \tag{4}$$

For example, the CPD for a faulty two-input NAND gate obtained using (4) is shown in Fig. 4(b). Note that all these products are obtained using the factor product as defined in the Appendix.

We motivate our formulation using an example. Fig. 5(a) and (b) shows the error model and the corresponding BN for reliability analysis of the circuit shown in Fig. 2(a). Since all the variables in this BN are binary-valued, we denote this BN model as the **two-valued BN**. As explained in Section II-B, the JPD of all the variables, $\boldsymbol{Z}$, in the BN can be written as a product of the CPDs of all the variables [using (1)]. For this example, therefore,

$$P(\mathbf{Z}) = P(X_1)P(X_2)P(X_3)f_1 f_2 f_3 P(E|Y,\hat{Y})$$

where

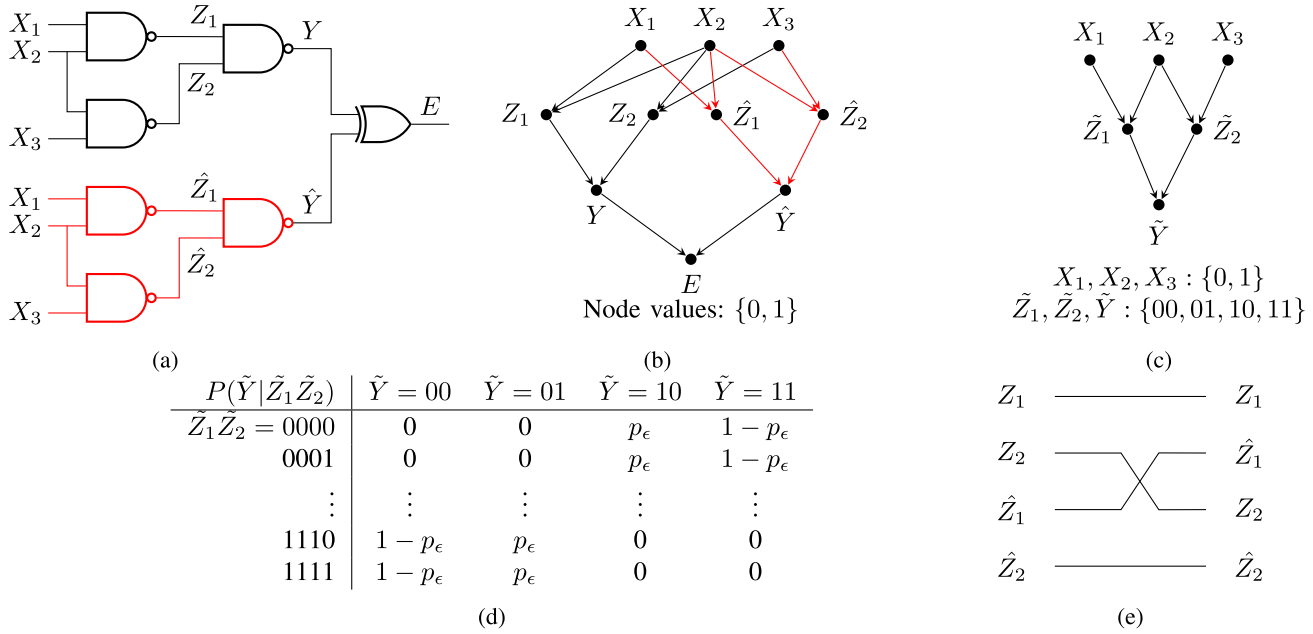
$$\begin{aligned} f_1 &= P(Z_1|X_1,X_2)P(\hat{Z}_1|X_1,X_2) = P(Z_1,\hat{Z}_1|X_1,X_2) \\ f_2 &= P(Z_2|X_2,X_3)P(\hat{Z}_2|X_2,X_3) = P(Z_2,\hat{Z}_2|X_2,X_3) \\ f_3 &= P(Y|Z_1,Z_2)P(\hat{Y}|\hat{Z}_1,\hat{Z}_2) = P(Y,\hat{Y}|Z_1,\hat{Z}_1,Z_2,\hat{Z}_2). \end{aligned} \tag{5}$$

This result follows from Property 1 for BNs. Here, for ease of explanation, the PIs ($\boldsymbol{X} = \{X_1, X_2, X_3\}$) are assumed to be error-free. The error probability of the circuit, $P(E)$, can be obtained by summing the JPD, $P(\mathbf{Z})$, over the states of all the other variables in the circuit, in a process called variable elimination [26], [37].

Each of the factors f_1, f_2, and f_3 in (5) can be considered to be a CPD, representing the JPD of the error-free and erroneous outputs given the values of the error-free and erroneous inputs. CPDs $P(Y|Z_1,Z_2)$ and $P(\hat{Y}|\hat{Z}_1,\hat{Z}_2)$ are 4×2 arrays, and therefore, the factor product will give $P(Y,\hat{Y}|Z_1,Z_2,\hat{Z}_1,\hat{Z}_2)$ as a 16×4 array.

Based on this grouping of CPDs, it can be easily seen that the system for reliability analysis can be represented as a single BN, with each variable $\{\tilde{Z}\}$ taking on four values $\{00, 01, 10, 11\}$ that denote the values of the corresponding signal in the fault-free and faulty circuits. This is shown in Fig. 5(c). We refer to the model in Fig. 5(c) as the **four-valued BN**. In the example, variables $\tilde{Z}_1$, $\tilde{Z}_2$, and $\tilde{Y}$ are four-valued. Since the PIs are assumed to be accurate for error rate computation, they are two-valued. In general, the PIs can also be four-valued. The CPD of each gate represents the JPD of the error-free and erroneous outputs, given the values of the error-free and erroneous inputs. Therefore,

$$P(\tilde{Y}|\tilde{Z}_1,\tilde{Z}_2) = P(Y,\hat{Y}|Z_1,\hat{Z}_1,Z_2,\hat{Z}_2). \tag{6}$$



$P(\tilde{Y}\|\tilde{Z}_1\tilde{Z}_2)$	$\tilde{Y}=00$	$\tilde{Y}=01$	$\tilde{Y}=10$	$\tilde{Y}=11$
$\tilde{Z}_1\tilde{Z}_2=0000$	0	0	p_ϵ	$1-p_\epsilon$
0001	0	0	p_ϵ	$1-p_\epsilon$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
1110	$1-p_\epsilon$	p_ϵ	0	0
1111	$1-p_\epsilon$	p_ϵ	0	0

(d)

(e)

Fig. 5. (a) Error model for computation of error rate for the circuit shown in Fig. 2(a). (b) Two-valued BN for the error model. (c) Four-valued BN for the error model. (d) CPD of a two-input NAND gate in the four-valued BN. (e) Change in variable order implemented using the permutation matrix.

B. CPDs in the Four-Valued BN

Practically, the CPD of a two-input gate in the four-valued BN, $P(\tilde{Y}|\tilde{Z}_1, \tilde{Z}_2)$, can be obtained as follows. Given $P(Y|Z_1, Z_2)$ and $P(\hat{Y}|\hat{Z}_1, \hat{Z}_2)$, we have

$$P(Y, \hat{Y}|Z_1, Z_2, \hat{Z}_1, \hat{Z}_2) = P(Y|Z_1, Z_2)P(\hat{Y}|\hat{Z}_1, \hat{Z}_2). \quad (7)$$

It is seen that CPDs in (6) and (7) differ only in the order of the input variables. Specifically, $\hat{Z}_1$ and Z_2 have to be interchanged as shown in Fig. 5(e). A change in the order of variables implies a permutation of the rows of the CPD. This can be done by premultiplying by a permutation matrix as follows:

$$P(\tilde{Y}|\tilde{Z}_1, \tilde{Z}_2) = P_m \times P(Y, \hat{Y}|Z_1, Z_2, \hat{Z}_1, \hat{Z}_2). \quad (8)$$

Here, P_m is a permutation matrix and $\times$ is used to represent matrix multiplication. Using the procedure described in [8], the permutation matrix can be written as follows:

$$\begin{aligned} P_m &= P(Z_1, \hat{Z}_1, Z_2, \hat{Z}_2|Z_1Z_2\hat{Z}_1\hat{Z}_2) \\ &= P(Z_1|Z_1)P(\hat{Z}_1, Z_2|Z_2, \hat{Z}_1)P(\hat{Z}_2|\hat{Z}_2) \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \end{aligned}$$

Here, $\otimes$ denotes the tensor product or Kronecker product of the two matrices. The conditional probabilities shown above are PTMs that are obtained using the approach described in [8]. The procedure for a gate with an arbitrary number of inputs is similar.

C. Computation of Error Rate and Reliability of the POs

The error rate and reliability of the POs of a circuit are defined as follows.

Definition 1 (Error Rate (p_{E_i})): It is the probability of error at a PO Y_i, averaged over all possible input values. It is assumed that all the inputs are equally likely. It is computed as follows:

$$p_{E_i} = P(\tilde{Y} = 01) + P(\tilde{Y} = 10). \quad (9)$$

Definition 2 (Reliability (r_{Y_i})): The reliability of an output Y_i is defined as $r_{Y_i} = 1 - p_{E_i}$.

The process of averaging over equally likely inputs is equivalent to computing the probability of error at the output after setting the signal probability of the PIs to 0.5.

As seen from (7) and (8), the CPD of net $\tilde{Y}$ in the four-valued formulations is simply a permutation of the factor product of the CPDs of the error-free net (Y) and the erroneous net ($\hat{Y}$) in the two-valued formulation. Therefore, the factor product of all CPDs in the four-valued will give the same overall JPD as the two-valued formulation. Exact computation of marginal probabilities $P(\tilde{Y})$ and $P(Y, \hat{Y})$ in the four and two-valued formulation involves summation of the JPD over the state of all other variables. Therefore, $P(\tilde{Y}) = P(Y, \hat{Y})$. Hence, the error rate [(9)] estimated using both the formulations is the same if *exact inference methods* are used.

The following propositions show that for approximate inference using sum-product BP, the four-valued BN is guaranteed to give an error rate of zero when the PIs and gates are error-free. The significance of this theorem is discussed after the proofs.

Definition 3: A net $\tilde{Z}_i$ is said to be error-free with respect to a probability distribution P if it satisfies $P(\tilde{Z}_i = 01) = P(\tilde{Z}_i = 10) = 0$.

Definition 4: The CPD of a net $\tilde{Y}$ with parents $Pa_{\tilde{Y}}$ is said to be error-free with respect to a distribution P if it satisfies the following:

$$P(\tilde{Y} = 01|Pa_{\tilde{Y}} = s) = P(\tilde{Y} = 10|Pa_{\tilde{Y}} = s) = 0$$

where

$$s \in S = \{\tilde{Z}_i \in \{00, 11\} \quad \forall\ \tilde{Z}_i \in Pa_{\tilde{Y}}\}.$$

Proposition 1: Given a set of error-free nets $\tilde{Z} = \{\tilde{Z}_1, \ldots, \tilde{Z}_n\}$, the joint probability $P(\tilde{Z})$ evaluates to zero if the state of any $Z_i \in \tilde{Z}$ is either 01 or 10.

Proof: The marginal probability of a net Z_i can be obtained from the joint probability $P(\tilde{Z})$ as follows: $P(\tilde{Z}_i) = \sum_{\tilde{Z} \setminus \tilde{Z}_i} P(\tilde{Z})$. Using Definition 3, $P(\tilde{Z}_i)$ evaluates to zero for states 01 and 10. Since probabilities are nonnegative, $P(\tilde{Z})$ should be zero for all the states such that $\tilde{Z}_i \in \{01, 10\}$. □

Proposition 2: If both the set of parents $Pa_{\tilde{Y}}$ and the CPD $P(\tilde{Y}|Pa_{\tilde{Y}})$ of a net $\tilde{Y}$ are error-free, $\tilde{Y}$ is also error-free.

Proof: Let S denote the set of all possible states of $Pa_{\tilde{Y}}$. S can be split into two disjoint sets S_1 and S_2, where S_1 comprises states where all the nets are either 00 or 11 (accurate states of the net) and $S_2 = S \setminus S_1$ (states where at least one net is inaccurate). The sum-product operations to find $P(\tilde{Y} = 01)$ can be written as follows:

$$P(\tilde{Y} = 01) = \sum_{s \in S_1} P(\tilde{Y} = 01|Pa_{\tilde{Y}} = s) P(Pa_{\tilde{Y}} = s) + \sum_{s \in S_2} P(\tilde{Y} = 01|Pa_{\tilde{Y}}) P(Pa_{\tilde{Y}} = s).$$

Since the CPD is error-free, the first term in the summation is zero. In the second term, the summation is over states of $Pa_{\tilde{Y}}$ such that at least one $\tilde{Z}_i \in Pa_{\tilde{Y}}$ is either 01 or 10. Using Proposition 1, the joint probability $P(Pa_{\tilde{Y}}) = 0$ for these states. Therefore, $P(\tilde{Y} = 01) = 0$. Similarly, $P(\tilde{Y} = 10) = 0$. □

Theorem 1: The output error rate estimated using sum-product BP in the four-valued BN is zero if all the PIs and gates are error-free.

Proof: In this proof, we use P to denote the exact CPDs present as initial factors and $Q(C)$ to denote the approximate JPD of a cluster C obtained after convergence of sum-product BP. Let $\tilde{Z}$ be a net in the circuit, $C_k = \{\tilde{Z}, Pa_{\tilde{Z}}, \tilde{R}\}$ be the cluster that is assigned the CPD of net $\tilde{Z}$ ($P(\tilde{Z}|Pa_{\tilde{Z}})$), and $Q_k(C_k)$ be the JPD of variables in C_k. Using (3), the marginal distribution $Q_k(\tilde{Z})$ can be written as follows:

$$\begin{aligned} Q_k(\tilde{Z}) &= \sum_{C_k \setminus \tilde{Z}} Q_k(C_k) = \sum_{C_k \setminus \tilde{Z}} \psi_i \prod_{j \in Neight(i)} m_{j \to i} \\ &= \sum_{Pa_{\tilde{Z}}, \tilde{R}} P(\tilde{Z}|Pa_{\tilde{Z}}) Q_k(Pa_{\tilde{Z}}) Q_k(\tilde{R}|\tilde{Z}, Pa_{\tilde{Z}}) \\ &= \sum_{Pa_{\tilde{Z}}} P(\tilde{Z}|Pa_{\tilde{Z}}) Q_k(Pa_{\tilde{Z}}). \end{aligned} \tag{10}$$

The second step in the equation follows from the chain rule of probability and because one of the initial factors assigned to C_k is the CPD of $\tilde{Z}$.

Let $\tilde{X} \in C_0$ be a PI, so that $Pa_{\tilde{X}} = \varnothing$ and the corresponding initial factor is $P(\tilde{X})$. Therefore, from (10), we have $Q_0(C_0) = P(\tilde{X})$. Consider a net $\tilde{Z} \in C_1$ that is driven by the PIs. As described in Section II, after convergence in sum-product BP, the marginal probability of a variable is the same in all the clusters in which it is present. Therefore, all the parents of $\tilde{Z}$ are also error-free with respect to Q_1. Since both the CPD and inputs in (10) are error-free, using Proposition 2, the marginal probability obtained after sum-product is also error-free. Thus, going to each cluster in topological order of the nets, all the nets are error-free with respect to the approximate distribution obtained after sum-product BP.

In particular, this is also true for POs. Therefore, the output error rate at all the POs evaluates to zero. □

TABLE I

OUTPUT ERROR RATE ESTIMATED USING LBP ON THE TWO-VALUED BN [FIG. 5(B)] AND FOUR-VALUED BN [FIG. 5(C)] FOR DIFFERENT GATE ERROR PROBABILITIES

p_ϵ	2-valued BN	4-valued BN
10^{-1}	0.501	0.213
10^{-3}	0.492	2.5×10^{-3}
10^{-6}	0.492	2.5×10^{-6}
0	0.492	0

There are several variants of sum-product BP that differ in the construction of the cluster graph used. In LBP and its variants, each node in the cluster graph contains a variable and its parents, and all the sep-sets contain a single variable. In GBP and IJGP, clusters and sep-sets can have larger sizes. IBIA uses sum-product BP in a sequence of approximate JTs. Since the underlying message-passing algorithm is the same, the estimates obtained with all these methods obey Theorem 1. In contrast, this is not guaranteed for the two-valued formulation. As a result, the estimated error rate using it does not scale well with gate error probabilities even for the small circuit in Fig. 5(a), as shown in Table I.

Corollary 1: The output error rate estimated after assuming independence between inputs of a gate in the four-valued formulation is zero if all PIs and gates are error-free.

Proof: In LBP, each sep-set contains a single variable and messages are in terms of marginals of variables. Each CPD is multiplied by the marginals of inputs to get the marginals of the output. This is the same as estimation of marginals assuming independence between inputs. Therefore, using Theorem 1, the estimated error rate is zero when the PIs and gates are error-free. □

Note that this can also be shown by traversing the nets in the topological order and applying Proposition 2.

D. Computation of Circuit Error Rate

Definition 5: Circuit Error Rate: (p_E) It is the probability of getting an error in at least one of the outputs. Equivalently, circuit reliability is $1 - p_E$.

p_E is computed after connecting a tree of OR gates to the POs, as shown in Fig. 1(c). We know that

$$\begin{aligned} p_E &= P(E_1 \vee E_2 \vee \cdots \vee E_m = 1) \\ &= 1 - P(E_1 = 0, E_2 = 0, \ldots, E_m = 0). \end{aligned} \tag{11}$$

Here, $\vee$ represents the OR operator. Therefore, the probability of getting an error in any one of the outputs can be computed from the probability that there is no error in any of the outputs. In the framework of BI, the joint probability of getting no error in any of the outputs can be computed

by setting $E_1 = E_2 = \cdots = E_m = 0$ as evidence states and computing the probability of evidence. The probability of evidence is referred to as the *partition function* in the BN literature. The partition function computation can be done using exact as well as many of the approximate BI algorithms. Note that LBP for computation of partition function will typically require a larger number of iterations for convergence and is not the same as assuming independence among the gate inputs.

To compute the required partition function using the four-valued formulation, we connect an additional dummy gate to each output with an associated CPD $P(E_i|\tilde{Y}_i)$ given by the following equation:

$P(E_i\|\tilde{Y}_i)$	$E_i = 0$	$E_i = 1$
$\tilde{Y}_i = 00$	1	0
01	0	1
10	0	1
11	1	0

The outputs of the dummy gates can be set as evidence states.

IV. Results

We evaluated the proposed formulation for reliability analysis by computing individual output error rates and the overall circuit error rate for various values of gate error probabilities. We used circuits belonging to two combinational benchmark suites, namely, ISCAS'85 [38] and the recent EPFL'15 [39] benchmarks. The benchmarks were synthesized using the Cadence Genus v15.2 tool using the Faraday 55-nm technology library. All the experiments were carried out on a 3.7-GHz Intel i7-8700 Linux system with 64-GB memory.

To compute the error rates of the outputs, the PIs are assumed to be error-free, with a signal probability of 0.5. However, the model itself supports any signal probability/reliability for the PIs. We have assumed that the error probabilities of all the gates are equal. Once again, the model supports any arbitrary gate error probabilities. As in all previous works [4], [16], [40], gates with fan-in greater than two are replaced by equivalent combinations of two-input gates. In the equivalent combination, all the gates other than the gate driving the final output are assumed to be error-free. For example, in a four-input AND gate replaced by a combination of three two-input AND gates, only the final AND gate is erroneous.

A. Baseline

To validate the proposed model, we use error rates obtained from logic simulation as the baseline. The simulation is carried out on the circuit configuration shown in Fig. 1(b), where the PIs feed both the ideal logic block (F) and the erroneous logic block ($\hat{F}$). Each gate in $\hat{F}$ has an independent input error signal with static probability p_ϵ as input. The corresponding outputs from F and $\hat{F}$ are XORed to determine the error rates (or equivalently the reliabilities) of the circuit outputs. The total number of inputs to the model is $N_{\text{PI}} + N_g$ where N_{PI} is the number of PIs and N_g is the number of gates in the circuit. The number of samples (N_s) used for the simulation must be large enough to ensure gate and output error probabilities are estimated well. The runtime required for the MC simulations is $O(N_g N_s)$.

For many of the large EPFL benchmarks (multiplier, div, and so on), Monte Carlo (MC) simulation with 10^6 was not possible since it ran out of memory. Therefore, we chose the simulation results obtained with 10^5 vectors as the baseline in all our evaluations. This implies that for output error rates of the order of 10^{-3}, the standard deviation $(((p(1-p))/10^5)^{1/2})$ in the estimated value is an order of magnitude lower. However, lower error rates cannot be estimated very reliably. Also for a gate error probability of 10^{-6}, this comparison was not possible, since at least 10^7 samples would be required.

TABLE II

Average Relative Error (in %) in Output Error Rate and Required Runtime for ISCAS'85 Benchmarks Using Different Deterministic Approximate Inference Techniques on the Four-Valued BN Formulation. The Maximum Cluster Size for IBIA, WMB, and IJGP Was Set to 10, and the Loop Depth for HAK Was Set to 3

	Avg. Relative Error (%)		Runtime (s)	
Methods	$\mathbf{p_\epsilon = 0.1}$	$\mathbf{p_\epsilon = 10^{-3}}$	$\mathbf{p_\epsilon = 0.1}$	$\mathbf{p_\epsilon = 10^{-3}}$
IBIA (10)	1.3	11.6	4	4
LBP*	3.8	25.9	0.006	0.006
HAK (LOOP3)*	4.1	44.4	259	468
IJGP $(10)^+$	27.1	46.5	386	385
WMB $(10)^+$	8.6	1750.4	620	619

*: Methods implemented in tool libDAI [41]
+: Methods implemented in tool Merlin [42]

B. Choice of Deterministic Inference Algorithms

We evaluated deterministic approximate BI techniques based on sum-product BP that are implemented in two publicly available tools, libDAI [41] and Merlin [42], and a more recent method IBIA [33] on the proposed BN formulation. Table II shows comparison of the required runtime and relative error in the error rate averaged across all POs and all ISCAS'85 benchmarks for different BI methods. The maximum cluster size was set to 10 for methods IBIA [33], WMB [32], and IJGP [31], and loop depth of 3 was set for HAK [43] which is the double-loop variant of GBP. The parameters were chosen based on runtime and memory constraints. In these methods, the runtime for inference is exponential in the maximum cluster size. For IBIA, it also depends on the number of junction trees constructed, while for HAK and IJGP, it depends on the number of iterations until convergence.

For a gate error probability of 0.1, the average relative error is less than 5% with IBIA, LBP, and HAK. On the other hand, for gate error probability of 10^{-3}, IBIA gives the least error followed by LBP. For WMB, the sum-product operation is approximated while computing messages and it is not guaranteed to satisfy Theorem 1. As seen from the table, it does not scale well with gate error probabilities.

For both the gate error probabilities, the runtime obtained with LBP is the least, followed by IBIA. It is seen from the table that the runtimes are relatively independent of the gate error probabilities. We have chosen LBP and IBIA

TABLE III

COMPARISON OF MAXIMUM ERROR AND RMSE OBTAINED USING IBIA ON THE TWO-VALUED AND FOUR-VALUED BNS FOR DIFFERENT GATE ERROR PROBABILITIES (p_ϵ). FOR $p_\epsilon = 0$, THE FOUR-VALUED BN GIVES AN ERROR RATE OF ZERO FOR ALL OUTPUTS. (A) MAXIMUM ERROR. (B) RMSE

(a)

	$p_\epsilon = 0.1$		$p_\epsilon = 0.001$		$p_\epsilon = 0$
	2-valued	4-valued	2-valued	4-valued	2-valued
c432	0.038	0.026	0.102	4.06-4	0.408
c499	0.007	0.007	0.124	0.001	0.5
c880	0.070	0.007	0.142	9.81×10^{-4}	0.23
c1355	0.002	0.006	0.007	7.71×10^{-4}	0.014
c1908	0.293	0.005	0.441	7.18×10^{-4}	0.5
c2670	0.060	0.003	0.489	0.003	0.5
c3540	0.076	0.037	0.489	0.057	0.5
c5315	0.086	0.032	0.491	0.004	0.5
c6288	0.022	0.004	0.453	0.177	0.5
c7552	0.084	0.033	0.490	0.014	0.5
Mean	**0.07**	**0.02**	**0.365**	**0.026**	**0.415**

(b)

	$p_\epsilon = 0.1$		$p_\epsilon = 0.001$		$p_\epsilon = 0$
	2-valued	4-valued	2-valued	4-valued	2-valued
c432	0.019	0.017	0.044	2.21×10^{-4}	0.172
c499	0.004	0.005	0.071	0.001	0.126
c880	0.015	0.002	0.044	3.72×10^{-4}	0.073
c1355	0.001	0.003	0.007	4.15×10^{-4}	0.008
c1908	0.055	0.002	0.261	2.82×10^{-4}	0.27
c2670	0.008	5.77×10^{-4}	0.075	2.89×10^{-4}	0.078
c3540	0.029	0.020	0.377	0.017	0.4
c5315	0.024	0.006	0.153	0.001	0.163
c6288	0.004	0.002	0.375	0.083	0.461
c7552	0.022	0.005	0.196	0.002	0.214
Mean	**0.024**	**0.006**	**0.166**	**0.01**	**0.196**

for evaluation in this work since these methods give good accuracies with reasonable runtimes.

C. Comparison With the Two-Valued BN Formulation

As mentioned previously, we refer to the model in Fig. 5(b) as the two-valued model and the model in Fig. 5(c) as the four-valued model. Table III has the maximum and root-mean-square (rms) error in the estimation of the output error rates for the ISCAS'85 benchmarks using the two-valued and four-valued formulations. Since errors obtained using the two-valued formulation were very large, we report absolute value of errors instead of relative errors. We have tabulated the results obtained with IBIA. Estimates obtained with LBP also follow a similar trend. The table has the results for gate error probabilities, $p_\epsilon = 0.1, 10^{-3}$ and 0. It is seen that the computation using the two-valued formulation does not scale well with the gate error probability (p_ϵ). For p_ϵ of 10^{-3}, the max-error for many outputs is close to 0.5, which is just noise. This is also true for $p_\epsilon = 0$. Both the max-error and RMSE are significantly lower when the four-valued formulation is used. The output error rates are zero when p_ϵ is 0, as guaranteed by Theorem 1. Fig. 6 shows the error rates at various outputs in the benchmark $c5315$ for different gate error probabilities and it is seen that it scales well with gate error probabilities.

As the gate error probability reduces, the correlation between the error-free and the corresponding erroneous signal increases. For discrete random variables, the mutual information (MI) between the two signals is a measure of this

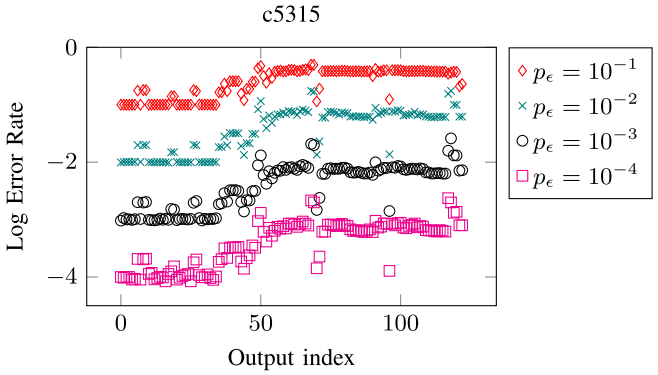


Fig. 6. Output error rate inferred using IBIA on the four-valued BN formulation for different gate error probabilities.

TABLE IV

AVERAGE MUTUAL INFORMATION BETWEEN THE ERROR-FREE AND THE CORRESPONDING ERRONEOUS NETS FOR VARIOUS GATE ERROR PROBABILITIES (p_ϵ)

	$p_\epsilon = 0.1$	$p_\epsilon = 10^{-3}$	$p_\epsilon = 10^{-6}$
c17	0.055	0.148	0.151
adder	0.088	0.205	0.21
alu_ctrl	0.046	0.154	0.159
dec	0.012	0.043	0.046
int2float	0.044	0.153	0.159
priority	0.064	0.133	0.136
router	0.049	0.089	0.091

correlation. The MI between a signal S and the corresponding erroneous signal $\hat{S}$ is defined as follows:

$$\mathrm{MI}_s = \sum_{S,\hat{S}} P(S,\hat{S}) \log \frac{P(S,\hat{S})}{P(S)P(\hat{S})}. \tag{12}$$

Table IV shows the average MI between the corresponding signals for some of small benchmarks, computed using IBIA. For these benchmarks, IBIA either performs exact inference or the error in the estimation is very small (of the order of 10^{-4}). It is clearly seen that the MI increases as the gate error probability reduces. This increased MI is taken care of in the four-valued formulation but the two-valued formulation struggles (and fails) to capture it. As seen from Table III, the estimation error obtained with the two-valued model increases drastically as the gate error probability is reduced.

D. Output Error Rate With the Four-Valued BN Model

Table V has the relative error in the computed error rate at the POs and the runtime for various benchmarks. The average (maximum) relative error is the average (maximum) of the absolute relative error over all the outputs. The percentage error in the corresponding reliability (r_{Y_i}) is shown in brackets. For IBIA, we used cluster sizes (i.e., the maximum number of variables in a cluster) of 7 and 10. The results are reported for two different gate error probabilities ($p_\epsilon = 0.1, 10^{-3}$). The table also shows the number of POs and the total number of gates obtained after replacing gates with fan-in greater than two with equivalent combinations of two-input gates.

Runtime: For both the values of p_ϵ, the same number of samples was used for logic simulation. As discussed, the runtime of IBIA and LBP is also relatively independent of gate error probabilities. So the table has only one set of runtimes.

TABLE V

STATISTICS OF BENCHMARKS USED FOR EVALUATION, AND TRADE-OFF BETWEEN RUNTIME (S) AND ACCURACY IN TERMS OF AVERAGE AND MAXIMUM RELATIVE ERROR (%) IN THE ERROR RATE (ER) USING IBIA AND LBP. THE RELATIVE ERROR IN RELIABILITY (R) IS INDICATED IN BRACKETS. RESULTS OBTAINED WITH LBP, IBIA USING MAX-CLUSTER SIZE OF 7 AND 10, AND MC SIMULATIONS ARE SHOWN IN COLUMNS MARKED AS "L," "I7," "I10," AND "MC," RESPECTIVELY. ISCAS'85 AND EPFL'15 CIRCUITS ARE SEPARATED BY A LINE

			Average Relative Error - ER (R) (%)						Maximum Relative Error - ER (R) (%)						Runtime (s)			
			$p_\epsilon = 0.1$			$p_\epsilon = 10^{-3}$			$p_\epsilon = 0.1$			$p_\epsilon = 10^{-3}$						
Circuit	**#Gates**	**#POs**	**L**	**I7**	**I10**	**L**	**I7**	**I10**	**L**	**I7**	**I10**	**L**	**I7**	**I10**	**L***	**I7***	**I10***	**MC+**
c432	182	7	13	4	4	30 (0.4)	3 (0.04)	1 (0.02)	40	8	7	61 (0.9)	7 (0.1)	3 (0.04)	0.03	0.4	0.9	16
c499	189	32	2	2	2	34 (0.1)	24 (0.08)	26 (0.08)	3	3	3	41 (0.1)	31 (0.1)	34 (0.1)	0.03	0.6	1.1	20
c1355	209	32	2	2	1.2	25 (0.1)	22 (0.07)	10 (0.03)	4	3	2	44 (0.1)	31 (0.1)	27 (0.08)	0.03	0.7	1.3	20
c1908	285	25	2	0.4	0.4	4 (0.05)	4 (0.03)	2 (0.02)	11	1.3	1.2	36 (0.6)	10 (0.2)	6 (0.07)	0.04	0.9	2	23
c880	308	26	2	0.9	0.5	8 (0.1)	4 (0.02)	4 (0.02)	8	4	2	27 (0.2)	23 (0.2)	18 (0.1)	0.05	0.8	2	24
c2670	613	139	1	0.8	0.1	2 (0.02)	1.5 (0.02)	0.7 (0.01)	20	22	0.8	27 (0.5)	25 (0.4)	15 (0.3)	0.1	1.2	2	52
c3540	930	22	6	5	4	25 (1)	24 (1)	23 (0.9)	35	38	35	78 (9)	75 (8)	74 (6)	0.1	4	8	53
c7552	1203	108	2	0.8	1	7 (0.1)	5 (0.1)	4 (0.08)	11	7	8	52 (3)	45 (2)	33 (2)	0.2	5	7	97
c5315	1232	123	2	1.4	1.2	8 (0.1)	6 (0.05)	4 (0.04)	14	21	19	42 (0.5)	50 (0.5)	48 (0.41)	0.2	4	8	79
c6288	2166	32	8	0.7	0.3	138 (12)	89 (11)	52 (7)	39	14	1.1	466 (28)	178 (30)	117 (21)	0.3	5	10	227
alu_ctrl	110	25	1.0	1.1	0.9	1 (0.01)	3 (0.01)	4 (0.01)	4	4	4	9 (0.04)	9 (0.04)	9 (0.04)	0.02	0.2	0.5	10
int2float	263	7	2	1.0	0.7	1 (0.01)	2 (0.01)	1 (0.01)	6	2	2	3 (0.01)	4 (0.03)	3 (0.03)	0.04	0.9	3	16
cavlc	825	11	9	9	9	14 (0.1)	13 (0.1)	14 (0.1)	30	30	30	49 (0.4)	49 (0.4)	49 (0.4)	0.1	7	17	41
i2c	1018	141	2	1.2	1.1	4 (0.02)	3 (0.01)	3 (0.01)	10	7	5	24 (0.1)	15 (0.08)	15 (0.07)	0.2	4	8	62
bar	2933	128	22	22	20	43 (0.7)	43 (0.7)	41 (0.7)	23	23	23	46 (0.8)	46 (0.8)	45 (0.8)	0.5	53	74	168
max	3413	130	8	2	2	100 (2)	31 (0.5)	15 (0.2)	11	7	5	150 (2)	74 (1)	33 (0.4)	0.5	40	24	303
arbiter	4938	129	2	2	2	8 (0.03)	7 (0.02)	6 (0.02)	7	7	6	19 (0.1)	18 (0.05)	17 (0.05)	1	86	188	288
sin	5179	25	0.3	0.4	0.4	59 (19)	59 (20)	54 (19)	2	2	2	128 (35)	114 (33)	107 (33)	1	88	111	520
voter	8369	1	2	2	2	107 (33)	74 (23)	53 (17)	2	2	2	107 (33)	74 (23)	53 (17)	1	25	30	749
square	17886	127	0.4	0.3	0.3	102 (21)	75 (16)	73 (15)	7	6	6	172 (34)	120 (30)	133 (29)	3	5528	1925	1616
multiplier	21875	128	1	0.3	0.3	100 (24)	100 (25)	89 (23)	5	5	5	239 (38)	192 (38)	195 (38)	4	1162	2404	2263
div	36218	128	2	1.1	1.1	37 (7)	22 (4)	21 (4)	31	13	10	266 (22)	89 (13)	117 (13)	6	447	683	3377
Mean			**4.2**	**2.7**	**2.4**	**39.0 (5.5)**	**27.9 (4.7)**	**22.2 (4.0)**	**14.7**	**10.3**	**8.2**	**94.8 (9.4)**	**58.2 (8.3)**	**52.4 (7.3)**				

*Runtimes with Python implementations for LBP, IBIA.
+Runtimes with Cadence Incisive v15.2; Number of vectors=10^5

As seen from the table, the runtime for both LBP and MC simulations is approximately linear in the number of gates, which is as expected. There are some deviations in MC simulations since we have used Cadence Incisive which is an event-driven simulator. The runtime of IBIA also increases with the number of gates. But the exact dependency is more difficult to predict. As mentioned previously, IBIA constructs a sequence of junction trees. As N_g increases, the number of junction trees also increases, but the exact number depends on the structure of the graph and the reconvergent loops in it. Due to this, there are some fluctuations in the runtimes.

For the same circuit, the runtime for IBIA increases with the maximum cluster size in nearly all the cases, as expected. For these cluster sizes, the time required is dominated by the time required to construct the junction trees rather than inference time. The exceptions occur because IBIA takes longer to construct a junction tree with the specified maximum cluster size.

The runtime for IBIA is much better than the MC simulations for most circuits and is comparable for a couple of large benchmarks (*square* and *multiplier*). However, if the gate error probability is reduced further, the runtime of IBIA and LBP will not change much, but the MC simulations will take significantly larger times since the number of samples required would be larger. For example, the average runtime over all the benchmarks for gate error probabilities of 10^{-1}, 10^{-3}, and 10^{-6} with IBIA is 276, 240, and 303 s, respectively. But full logic simulation is not possible for $p_\epsilon = 10^{-6}$.

Accuracy: For $p_\epsilon = 0.1$, the mean maximum relative error reduces from 15% with LBP to 8% with IBIA using a cluster size of 10. However, the mean average relative error obtained with all three methods is small (2%–4%). Therefore, for this and larger gate error probabilities reasonably accurate estimates can be obtained within very small runtimes if we assume independence between inputs of a gate.

When the gate error probabilities are reduced to 10^{-3}, the relative error in the estimates of the error rates is larger, with IBIA giving significantly better estimates than LBP with a mean max-relative error of 52% and 95%, respectively. For some large benchmarks in EPFL arithmetic circuits, the maximum relative error is greater than 100%. However, the relative error in the corresponding reliabilities (shown in brackets) is much lower. This is because for small gate error probabilities, the output error rate is small and the corresponding reliability is large. Therefore, though the absolute error in both the reliability and error rate is the same, the percentage error with respect to reliability is much smaller. We have reported both since some of the earlier works report results with error rates, while others for reliability. On an average, the maximum and average relative error in reliability obtained with IBIA is about 7% and 4%, respectively. As expected, for both the gate error probabilities, we observe that the accuracy improves as the cluster size increases.

Among the existing methods [4], [23], [24], [25], [44], the results have been reported for ISCAS'85 benchmarks with gate error probabilities of 0.01 or larger. We have not seen any results for large EPFL benchmarks for any gate error probabilities. A fair comparison of results obtained with the existing methods is difficult since the synthesized netlists may vary in each case. However, just to compare the overall trend, we observe that the average relative error in the reliability of POs reported with these methods is between 1.2% and 3.8%

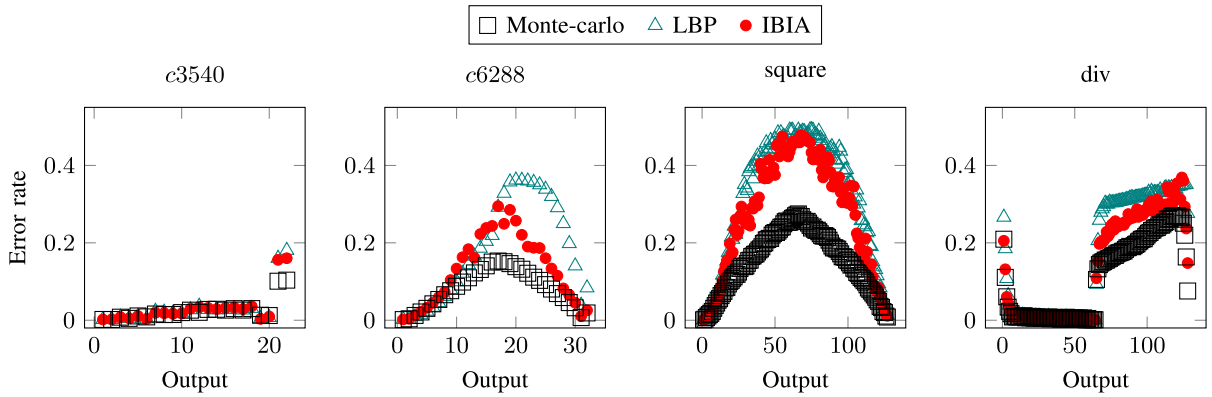


Fig. 7. Error rate at the POs obtained using MC simulations, IBIA, and LBP for $p_\epsilon = 10^{-3}$.

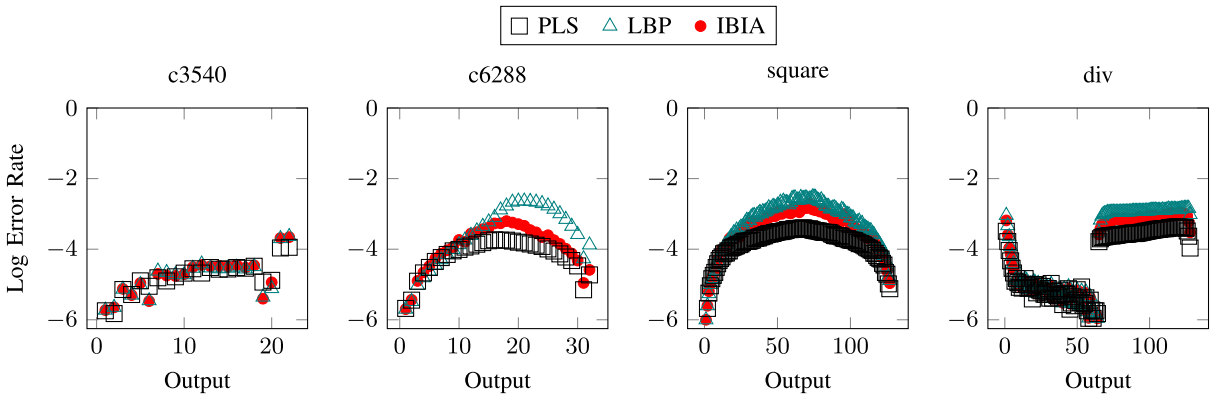


Fig. 8. Error rate at the POs obtained using PLS, IBIA, and LBP for $p_\epsilon = 10^{-6}$.

for gate error probabilities of 0.1 and 0.01. In contrast, the average error in reliability with the proposed formulation is 0.7% with IBIA (cluster size = 10) and 2.5% if we assume independence between inputs of a gate. Very few results are available for gate error probabilities $p_\epsilon \leq 0.001$. Both [4] and [20] report errors averaged across several gate error probabilities. Thus, a direct comparison is not possible. In [20], with zero gate error probabilities, the relative error in the error rate is 45% for a small circuit like $c2670$. In contrast, we show that our method is guaranteed to give zero error rates if algorithms based on sum-product BP (such as LBP, GBP, and IBIA) are used for inference.

Fig. 7 shows the computed error rates at each PO using MC, IBIA (with cluster size = 10), and LBP for some of the benchmarks in which the percentage error in the estimated error rate is large. It is seen that while IBIA and LBP overestimate the error rates, they are able to capture the error rates of outputs relative to each other quite well in most cases. In some testcases like div and $c6288$, the accuracy improves when larger clusters are used. The accuracy obtained with IBIA is comparable to LBP for square and $c3540$. Since the relative error rate among the outputs is approximately preserved by both the inference methods (with greater accuracy by IBIA than LBP), these can be used to identify the outputs that are more susceptible to error. This will enable targeted application of techniques to improve the reliability of the circuit.

Fig. 8 shows the logarithm of the error rates at the POs obtained using LBP and IBIA when the gate error probability is further reduced to $p_\epsilon = 10^{-6}$. The MC simulations were not possible for this gate error probability due to memory errors. Therefore, we used PLS [14], a sampling-based approximate BI technique. The implementation of PLS was taken from the SMILE toolkit [45], and the number of samples was set to 10^7. It is seen that the estimated error rate scales well when the gate error probability is reduced. Once again, the relative reliability among the outputs is captured well by both LBP and IBIA. The error rates obtained using LBP are consistently larger than with IBIA. While PLS is possibly more accurate, the runtime is about an order of magnitude larger than IBIA. For example, benchmark div requires 440 min with PLS, while the runtime with IBIA is only 11 min.

E. Circuit Error Rate With the Four-Valued BN Model

Table VI shows the relative error in the circuit error rate (Definition 5) for $p_\epsilon = 10^{-3}$. It also shows the estimates obtained after the MC simulations on the circuit configuration shown in Fig. 1(c). For $p_\epsilon = 0.1$, the circuit error rate becomes one and both the methods give close to accurate estimates. The table has a comparison of the two methods used, namely, (a) connecting a tree of OR gates and (b) using partition function computation [see (11)]. It is seen that the relative error in the estimation is significantly lower when PR formulation is used for both the methods, especially for the smaller ISCAS'85 benchmarks. Although the average over all the benchmarks is comparable, it is seen that LBP

TABLE VI
CIRCUIT ERROR RATE (p_E) OBTAINED USING MC SIMULATIONS, THE REQUIRED RUNTIME (S), AND THE RELATIVE ERROR (IN %) IN THE CIRCUIT ERROR RATE OBTAINED USING TWO INFERENCE METHODS. IBIA/LBP-OR REFERS TO COMPUTATION AFTER CONNECTING A TREE OF OR GATES AND IBIA/LBP-PR REFERS TO COMPUTATION USING THE PARTITION FUNCTION

	p_E	Relative Error (%)				Runtime (s)	
	MC	IBIA-OR	IBIA-PR	LBP-OR	LBP-PR	LBP-PR[1]	IBIA-PR[2]
c432	0.041	32	4	169	23	0.02	1
c499	0.087	21	18	47	5	0.02	1
c880	0.094	11	5	25	3	0.04	2
c1355	0.085	6	5	40	4	0.02	1
c1908	0.101	35	15	62	4	0.04	2
c2670	0.210	21	0.3	46	2	0.1	3
c3540	0.186	53	27	179	2	0.2	10
c5315	0.329	40	7	45	1	0.2	8
c6288	0.778	23	21	28	34	0.8	11
c7552	0.339	43	23	77	3	0.2	7
alu_ctrl	0.059	23	1	46	2	0.01	1
arbiter	0.352	7	7	9	5	0.5	172
cavlc	0.164	6	7	7	5	0.1	18
i2c	0.317	33	12	44	1	0.1	7
int2float	0.046	8	4	9	1	0.03	3
voter	0.238	53	53	107	104	12	31
bar	0.679	0	9	2	31	1	71
max	0.263	175	97	275	5	1	35
sin	0.864	16	16	16	0.3	3	110
div	0.978	2	2	2	9	224	775
square	1.000	0	0	0	0	6	3526
multiplier	1.000	0	0	0	0	14	5516
Mean		**27.6**	**15.2**	**56.1**	**11.1**	**12**	**469**

[1]Runtime with C++ implementation of LBP from libDAI
[2]Runtime with Python implementation of IBIA

gives significantly lower relative errors for a large number of benchmarks. For testcases *square* and *multiplier*, the circuit error rate is very close to one and both the methods perform well. Though LBP requires more iterations to converge when the PR formulation is used, it is quite fast, thus making it suitable for use in an optimization framework. The runtime for IBIA is expected to be larger since it uses larger cluster sizes. That said, the reported runtimes for LBP and IBIA are not directly comparable since both are implemented in different programming languages. The average runtime for LBP and IBIA for $p_\epsilon = 0.1$ is 17 and 377 s, respectively, which is similar to the runtimes for $p_\epsilon = 0.001$.

V. COMPARISON WITH RELATED WORK

We compare our approach with the existing methods that either use (a) a four-valued formulation or (b) BI techniques for reliability estimation.

Several existing approaches [4], [15], [20], [21], [23], [24], [25] use a single copy of the circuit with additional probabilities for each net. The problem here is the accurate propagation of these probabilities in the presence of reconvergent fanouts. Quick estimates can be obtained if gate inputs are assumed to be independent [15], [21]. However, as seen from the results, the accuracy of this method drops as the gate error probability reduces. Methods in [4] and [20] use an extension of the correlation coefficient method (CCM) to compute signal correlation coefficients. In [23], [24], and [25], correlation coefficients are computed with respect to signal reliabilities. These correlations are estimated using analytical methods [4], [16], [23] or simulation-based methods [20], or hybrid methods that combine these two approaches [46]. Typically, pairwise correlations are computed which limits the accuracy of these methods. In [20] and [46], the accuracy is limited by the length of the bitstreams used to estimate the correlation coefficients. The methods proposed in [4], [23], [24], [25], and [44] require the computation of signal probabilities in error-free circuits to estimate correlations in reliability. This itself is a #P-complete problem. Accurate estimates obtained using BDDs have been used in [4]. However, this limits the scalability of the method to relatively small circuits. In contrast, our method does not require these probabilities. We avoid inaccuracies in the estimation of correlation coefficients by deriving CPDs corresponding to the four-valued signals and using BI techniques that give approximate joint distribution over larger clusters of variables. Unlike the existing approaches, in our approach, it is possible to tradeoff runtime and accuracy by increasing the cluster sizes.

The BI techniques for reliability analysis have been used in [5], [11], [12], [47], and [48]. In [5], the two-valued formulation has been used along with the sampling-based approximate inference techniques. As with all the sampling methods, this approach is inflexible in the sense that any change in the circuit requires a complete reevaluation and it becomes expensive as the gate error probability reduces. Since the time complexity for sampling techniques scales linearly with the number of nodes in the network, it can be reduced to half using the proposed four-valued BN instead of the two-valued formulation. Exact BI methods have also been used for the estimation of gate error probabilities based on device-level parameters [11], [47], and for the computation of bounds on reliability by identifying the worst case input vector [12], [48]. However, exact inference is possible only for small circuits.

VI. CONCLUSION

We propose a novel algorithm for the estimation of error rate/reliability in probabilistic and unreliable circuits. Our method scales well with gate error probabilities and preserves the relative reliability of the outputs. We also propose a novel method for computing the overall circuit error rate by casting it as a problem of estimation of partition function in BNs. This formulation gives good accuracies within reasonable runtimes, making it suitable for use in an optimization framework. Although we have demonstrated results for CMOS circuits, the methods proposed are general and can be used for circuits that are built with post-CMOS devices. They can also be used for the analysis and design of approximate circuits. For these circuits, the four-valued formulation can be applied directly to the accurate and imprecise truth tables, since the error rate is independent of the implementation.

APPENDIX

A conditional distribution $P(\mathbf{Y}|\mathbf{X})$ is a *factor* ϕ over the set of variables $\mathbf{Y} \cup \mathbf{X}$. Let $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ be disjoint sets of variables and $\phi_1(\mathbf{X}, \mathbf{Y}), \phi_2(\mathbf{Y}, \mathbf{Z})$ be two factors. The *factor product* [26, Ch. 4] $\phi_1\phi_2$ gives a factor ψ which is obtained as follows:

$$\forall x, y, z \in \text{Domain}(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$$
$$\psi(\mathbf{X}, \mathbf{Y}, \mathbf{Z} = x, y, z) = \phi_1(x, y)\phi_2(y, z).$$

If factors ϕ_1 and ϕ_2 contain disjoint sets of variables, then the factor product is the same as the tensor product.

REFERENCES

[1] J. Von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," *Automata Stud.*, vol. 34, pp. 43–98, Dec. 1956.

[2] L. Tan, Z. Li, G. Su, and D. Wang, "Asymptotically linear analysis and gate probability allocation schemes in probabilistic circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 2, pp. 596–606, Feb. 2020.

[3] N.-Z. Lee and J.-H.-R. Jiang, "Towards formal evaluation and verification of probabilistic design," *IEEE Trans. Comput.*, vol. 67, no. 8, pp. 1202–1216, Aug. 2018.

[4] M. R. Choudhury and K. Mohanram, "Reliability analysis of logic circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 3, pp. 392–405, Mar. 2009.

[5] T. Rejimon, K. Lingasubramanian, and S. Bhanja, "Probabilistic error modeling for nano-domain logic circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 17, no. 1, pp. 55–65, Jan. 2009.

[6] B. Krishnamurthy and I. G. Tollis, "Improved techniques for estimating signal probabilities," *IEEE Trans. Comput.*, vol. 38, no. 7, pp. 1041–1045, Jul. 1989.

[7] S. Krishnaswamy, G. F. Viamontes, I. L. Markov, and J. P. Hayes, "Accurate reliability evaluation and enhancement via probabilistic transfer matrices," in *Proc. Design, Autom. Test Eur.*, Mar. 2005, pp. 282–287.

[8] S. Krishnaswamy, G. F. Viamontes, I. L. Markov, and J. P. Hayes, "Probabilistic transfer matrices in symbolic reliability analysis of logic circuits," *ACM Trans. Des. Automat. Electron. Syst.*, vol. 13, no. 1, pp. 1–35, 2008.

[9] J. Han, H. Chen, E. Boykin, and J. Fortes, "Reliability evaluation of logic circuits using probabilistic gate models," *Microelectron. Rel.*, vol. 51, no. 2, pp. 468–476, 2011.

[10] O. Keszocze, "BDD-based error metric analysis, computation and optimization," *IEEE Access*, vol. 10, pp. 14013–14028, 2022.

[11] W. Ibrahim, V. Beiu, and A. Beg, "GREDA: A fast and more accurate gate reliability EDA tool," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 31, no. 4, pp. 509–521, Apr. 2012.

[12] W. Ibrahim, M. Shousha, and J. W. Chinneck, "Accurate and efficient estimation of logic circuits reliability bounds," *IEEE Trans. Comput.*, vol. 64, no. 5, pp. 1217–1229, May 2015.

[13] J. Han, H. Chen, J. Liang, P. Zhu, Z. Yang, and F. Lombardi, "A stochastic computational approach for accurate and efficient reliability evaluation," *IEEE Trans. Comput.*, vol. 63, no. 6, pp. 1336–1350, Jun. 2014.

[14] M. Henrion, "Propagating uncertainty in Bayesian networks by probabilistic logic sampling," in *Uncertainty in Artificial Intelligence* (Machine Intelligence & Pattern Recognition), vol. 5, J. F. Lemmer and L. N. Kanal, Eds. Amsterdam, The Netherlands: North Holland, 1988, pp. 149–163.

[15] D. T. Franco, M. C. Vasconcelos, L. Naviner, and J.-F. Naviner, "Reliability analysis of logic circuits based on signal probability," in *Proc. 15th IEEE Int. Conf. Electron., Circuits Syst.*, Aug. 2008, pp. 670–673.

[16] S. Ercolani, M. Favalli, M. Damiani, P. Olivo, and B. Ricco, "Estimate of signal probability in combinational logic networks," in *Proc. 1st Eur. Test Conf.*, Jan. 1989, pp. 132–133.

[17] S. Ercolani, M. Favalli, M. Damiani, P. Olivo, and B. Ricco, "Testability measures in pseudorandom testing," *IEEE Trans. Comput.-Aided Design Integr.*, vol. 11, no. 6, pp. 794–800, Jun. 1992.

[18] S. Sivaswamy, K. Bazargan, and M. Riedel, "Estimation and optimization of reliability of noisy digital circuits," in *Proc. 10th Int. Symp. Quality Electron. Design*, Mar. 2009, pp. 213–219.

[19] N. Mohyuddin, E. Pakbaznia, and M. Pedram, "Probabilistic error propagation in a logic circuit using the Boolean difference calculus," in *Advanced Techniques in Logic Synthesis, Optimizations and Applications*. Cham, Switzerland: Springer, 2011, pp. 359–381.

[20] H. Jahanirad, "CC-SPRA: Correlation coefficients approach for signal probability-based reliability analysis," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 4, pp. 927–939, Apr. 2019.

[21] D. T. Franco, M. C. Vasconcelos, L. Naviner, and J.-F. Naviner, "Reliability of logic circuits under multiple simultaneous faults," in *Proc. 51st Midwest Symp. Circuits Syst.*, Aug. 2008, pp. 265–268.

[22] J. T. Flaquer, J. M. Daveau, L. Naviner, and P. Roche, "Fast reliability analysis of combinatorial logic circuits using conditional probabilities," *Microelectron. Rel.*, vol. 50, pp. 1215–1218, Sep./Nov. 2010.

[23] C. Chen and R. Xiao, "A fast model for analysis and improvement of gate-level circuit reliability," *Integration*, vol. 50, pp. 107–115, Jun. 2015.

[24] J. Cai and C. Chen, "Circuit reliability analysis using signal reliability correlations," in *Proc. IEEE Int. Conf. Softw. Qual., Rel. Secur. Companion (QRS-C)*, Jul. 2017, pp. 171–176.

[25] K. Sikander, S. Zhan, and C. Chen, "An analytical model for circuit reliability estimation," in *Proc. IEEE Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2021, pp. 84–87.

[26] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA, USA: MIT Press, 2009.

[27] K. P. Murphy, Y. Weiss, and M. I. Jordan, "Loopy belief propagation for approximate inference: An empirical study," in *Uncertainty in Artificial Intelligence*. Amsterdam, The Netherlands: Elsevier, 1999, pp. 467–475.

[28] W. Wiegerinck and T. Heskes, "Fractional belief propagation," in *Advances in Neural Information Processing Systems*, vol. 15. Cambridge, MA, USA: MIT Press, 2003.

[29] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, "Tree-reweighted belief propagation algorithms and approximate ML estimation by pseudo-moment matching," in *Proc. Int. Workshop Artif. Intell. Statist.*, 2003, pp. 308–315.

[30] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Generalized belief propagation," in *Proc. NIPS*, vol. 13, 2000, pp. 689–695.

[31] R. Mateescu, K. Kask, V. Gogate, and R. Dechter, "Join-graph propagation algorithms," *J. Artif. Intell. Res.*, vol. 37, pp. 279–328, Mar. 2010.

[32] Q. Liu and A. Ihler, "Bounding the partition function using holder's inequality," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 849–856.

[33] S. Bathla and V. Vasudevan, "IBIA: Bayesian inference via incremental build-infer-approximate operations on clique trees," 2022, *arXiv:2202.12003*.

[34] M. Chavira and A. Darwiche, "On probabilistic inference by weighted model counting," *Artif. Intell.*, vol. 172, nos. 6–7, pp. 772–799, Apr. 2008.

[35] C. Yuan and M. J. Druzdzel, "An importance sampling algorithm based on evidence pre-propagation," in *Uncertainty in Artificial Intelligence*. Amsterdam, The Netherlands: Elsevier, 2002, pp. 624–631.

[36] A. E. Gelfand, "Gibbs sampling," *J. Amer. Stat. Assoc.*, vol. 95, no. 452, pp. 1300–1304, Dec. 2000.

[37] N. L. Zhang and D. Poole, "Exploiting causal independence in Bayesian network inference," *J. Artif. Intell. Res.*, vol. 5, pp. 301–328, Dec. 1996.

[38] F. Brglez, "A neural netlist of 10 combinational benchmark circuits," *Proc. IEEE Special Session ATPG Fault Simulation*, Jun. 1985, pp. 151–158.

[39] L. Amarú, P.-E. Gaillardon, and G. De Micheli, "The EPFL combinational benchmark suite," in *Proc. Int. Workshop Log. Synth. (IWLS)*, 2015, pp. 1–5.

[40] S. Bhanja and N. Ranganathan, "Cascaded Bayesian inferencing for switching activity estimation with correlated inputs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 12, no. 12, pp. 1360–1370, Dec. 2004.

[41] J. M. Mooij, "libDAI: A free and open source C++ library for discrete approximate inference in graphical models," *J. Mach. Learn. Res.*, vol. 11, pp. 2169–2173, Aug. 2010.

[42] R. Marinescu. (2016). *Merlin*. Accessed: Oct. 15, 2021. [Online]. Available: https://github.com/radum2275/merlin/

[43] T. Heskes, K. Albers, and B. Kappen, "Approximate inference and constrained optimization," in *Uncertainty in Artificial Intelligence*. Amsterdam, The Netherlands: Elsevier, 2003, pp. 313–320.

[44] C. Chen, J. Cai, and S. Zhan, "A triple-point model for circuit-level reliability analysis," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2018, pp. 1–4.

[45] M. J. Druzdzel, "Smile: Structural modeling, inference, and learning engine and genie: A development environment for graphical decision-theoretic models," in *Proc. AAAI Conf. Artif. Intell.*, 1999, pp. 902–903.

[46] S. Zhan and C. Chen, "A hybrid method for signal probability and reliability estimation with combinational circuits," *Integration*, vol. 87, pp. 275–283, Nov. 2022.

[47] W. Ibrahim and V. Beiu, "Using Bayesian networks to accurately calculate the reliability of complementary metal oxide semiconductor gates," *IEEE Trans. Rel.*, vol. 60, no. 3, pp. 538–549, Sep. 2011.

[48] W. Ibrahim and H. Ibrahim, "Multithreaded and reconvergent aware algorithms for accurate digital circuits reliability estimation," *IEEE Trans. Rel.*, vol. 68, no. 2, pp. 514–525, Jun. 2019.