Microelectronics Reliability xxx (2010) xxx–xxx

Contents lists available at ScienceDirect

Microelectronics Reliability

journal homepage: www.elsevier.com/locate/microrel

Reliability evaluation of logic circuits using probabilistic gate models

Jie Han [a,*], Hao Chen [a], Erin Boykin [b], José Fortes [b]

[a] Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Alberta, Canada T6G 2V4
[b] Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA

ARTICLE INFO

Article history:
Received 31 January 2010
Received in revised form 28 July 2010
Accepted 28 July 2010
Available online xxxx

ABSTRACT

Logic circuits built using nanoscale technologies have significant reliability limitations due to fundamental physical and manufacturing constraints of their constituent devices. This paper presents a probabilistic gate model (PGM), which relates the output probability to the error and input probabilities of an unreliable logic gate. The PGM is used to obtain computational algorithms, one being approximate and the other accurate, for the evaluation of circuit reliability. The complexity of the approximate algorithm, which does not consider dependencies among signals, increases linearly with the number of gates in a circuit. The accurate algorithm, which accounts for signal dependencies due to reconvergent fanouts and/or correlated inputs, has a worst-case complexity that is exponential in the numbers of dependent reconvergent fanouts and correlated inputs. By leveraging the fact that many large circuits consist of common logic modules, a modular approach that hierarchically decomposes a circuit into smaller modules and subsequently applies the accurate PGM algorithm to each module, is further proposed. Simulation results are presented for applications on the LGSynth91 and ISCAS85 benchmark circuits. It is shown that the modular PGM approach provides highly accurate results with a moderate computational complexity. It can further be embedded into an early design flow and is scalable for use in the reliability evaluation of large circuits.

1. Introduction

As CMOS technology enters the nanometer regime, shrinking device dimensions, lower design tolerances and fabrication variability have negative impacts on reliability and result in increased device failure rates [1]. The effects of process variations, due to random dopant fluctuations or sub-wavelength lithography, are expected to reduce transistor reliability as technology further scales. Permanent faults can be caused by time-dependent dielectric breakdown of materials, hot carrier injection effects and negative bias temperature instability in transistors [2]. Electromigration becomes a major concern for interconnect reliability and can lead to faults due to connection shorts and opens. Furthermore, transient (soft) errors may result from temporary environmental influences [3]. Higher integration densities and lower voltage/current thresholds have increased soft error rates in VLSI circuits.

Non-conventional nanotechnologies, currently being investigated as potential alternatives to CMOS, are expected to have lower reliability than current CMOS technology. This is a result of manufacturing processes and sensitivity to environmental factors – nondeterministic behaviors will be present due to quantum effects, environmental noise and, in some cases, inexpensive but inaccurate chemical self-assembly [4]. The imprecision and randomness inherent to the stochastic nature of chemical self-assembly will inevitably raise the density of defects in molecular devices, which subsequently cause malfunctions of logic gates and interconnects in circuits. The reliability limitations of nanoscale devices have become first-order issues and probabilistic designs, rather than deterministic ones, will be necessary to account for the stochastic behavior of nanoscale circuits and systems [5].

The design of "probabilistic logics" has been of interest since the early days of electronic computers when von Neumann proposed to synthesize reliable systems from unreliable components [6]. In his study, errors are treated probabilistically and a system is considered reliable if the probability of its correct output is greater than a threshold. As von Neumann stated, when the probability of output error reaches this threshold, the results from computation become irrelevant to the inputs and restoration of the outputs to their correct signal values is not possible. von Neumann's work has motivated many efforts to characterize the reliability of fault-tolerant architectures in both conventional [7] and nanotechnology [8] systems. In light of the continuous scaling of CMOS and the emergence of new nanoscale technologies, reliability has increasingly been a concern and is expected to become a major design metric as performance and power are for today. This increasing demand on reliability design calls for accurate and efficient evaluation tools for the analysis of circuit reliability. Reliability evaluation through analysis and/or simulation also serves as the first step towards understanding when fault-tolerance needs to be added to a system and what the resulting reliability gains are.

* Corresponding author. Tel.: +1 780 492 1361; fax: +1 780 492 1811.
E-mail address: jhan8@ualberta.ca (J. Han).

0026-2714/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.microrel.2010.07.154

Please cite this article in press as: Han J et al. Reliability evaluation of logic circuits using probabilistic gate models. Microelectron Reliab (2010), doi:10.1016/j.microrel.2010.07.154

To this end, various approaches based on probabilistic analysis have been proposed for the evaluation of circuit reliability [9,10]. Recent researches have focused on the use of Markov random fields [11], probabilistic model checking (PMC) [12], probabilistic transfer matrices (PTMs) [13], Bayesian networks [14], analytical and scalable approaches [15], probabilistic decision diagrams (PDDs) [16], Boolean difference calculus [17], signal probabilities [18], circuit transformations [19] and multiple passes for sequential circuits [20]. Several of these approaches, such as those using PMC [12], PTMs [13] and PDDs [16], provide accurate evaluation results, however, they are affected by the problems of state space explosion and an exponential complexity, which make them practically infeasible to be used for large circuits. While the other techniques can be more efficient in terms of runtime or memory usage [14,15,17–19], they generally provide approximate results. In [14], an approximate inference scheme is proposed for the handling of large circuits using a probabilistic model based on Bayesian networks. A novel approach using Boolean difference calculus is applied to the probabilistic analysis of logic circuits in [17]. In [18], several algorithms based on the straightforward application of signal probabilities are presented to estimate circuit reliability. In [19], the signal probabilities of all internal nodes of logic circuits are calculated using techniques of circuit transformation. In [15], three scalable algorithms are proposed for reliability analysis. Particularly, a so-called single-pass method is able to accurately evaluate circuits without reconvergent fanouts and is applicable to other circuits by computing the correlation coefficients of dependent signals. This single-pass method is extended in [20] in the form of multiple passes for the reliability evaluation of sequential circuits.

In this paper, we present computational algorithms using probabilistic gate models (PGMs) for the reliability analysis of logic circuits. A simple algorithm, directly applicable on the interconnections of gates as specified in a circuit's netlist, provides highly accurate results when the circuit has no or few reconvergent fanout, as well as when the fanouts originate from the primary inputs and the primary inputs are highly reliable. An accurate algorithm is able to determine the exact reliability of a circuit, while in the worst case, its complexity increases exponentially with the number of dependent reconvergent fanouts. Based on these algorithms, a modular approach is further investigated to explore the tradeoff between the speed and the accuracy in reliability evaluation. It is shown that the hierarchical modular approach can be embedded into a design flow and is potentially useful for the reliability evaluation of VLSI circuits and systems. Incorporated with various gate characteristics, these algorithms can be used to evaluate the necessity of using fault mitigation techniques on circuit and logic levels. Although it is illustrated using non-redundant circuits, the proposed methodology is general and thus also applicable to circuits where redundancy and voting are used for fault-tolerance purposes.

This paper is organized as follows. In Section 2, the notions of PGMs, as well as a general procedure to produce PGMs, are presented. Section 3 presents PGM-based analytical approaches to reliability modeling and reports computational results of applying them to benchmark circuits. Section 4 investigates a modular approach that trades off accuracy for efficiency in evaluating reliability. Section 5 concludes the paper.

2. Probabilistic gate models for unreliable logic gates

2.1. Probabilistic gate models (PGMs)

We follow the definitions in what von Neumann called "probabilistic logics," where Boolean signals are considered probabilistic and a logic gate is assumed to fail independently with a constant probability ε [6,21–24]. A PGM is based on such a model of unreliable gates. While simplistic, this error model can be extended to consider technologies used to build gates by incorporating their fault mechanisms and gate structures. For instance, one could develop more accurate models based on a gate's transistor-level structure [25,26], its physical area or its robustness to errors, and then incorporate these factors into the expression of ε.

We further assume that each binary signal of a gate's input or output is associated with a random variable, which denotes the probability of this signal, defined as follows:

Definition 1. The signal probability of an input or output of a gate is defined as the probability that the signal is a logical "1."

Given a gate error rate ε, it then becomes possible to relate a gate's output probability to its input probabilities, according to the function and malfunction of the gate. Given an output's signal probability, $P(\text{“1”})$, and its complement, $P(\text{“0”}) = 1 - P(\text{“1”})$, the reliability of the output is given by:

$$R = P(\text{“1”})^{*}P_e(\text{“1”}) + P(\text{“0”})^{*}P_e(\text{“0”}), \tag{1}$$

where $P_e(\text{“1”})$ and $P_e(\text{“0”})$ are the probabilities that the output is expected to be a "1" and "0" respectively, provided that the gate is fault-free.

If the output is expected to be a "1," i.e. $P_e(\text{“1”}) = 1$ and $P_e(\text{“0”}) = 0$, Eq. (1) reduces to $R = P(\text{“1”})$, i.e., the obtained probability is the reliability of the output. If the output is expected to be a "0," i.e. $P_e(\text{“1”}) = 0$ and $P_e(\text{“0”}) = 1$, Eq. (1) reduces to $R = P(\text{“0”}) = 1 - P(\text{“1”})$, i.e., the output reliability is the complement of the obtained probability.

For convenience, we shall mean "the probability (of a signal) being a "1" by using the simple term "probability" throughout the text, unless it is otherwise noted.

The PGM for a NAND gate is given as follows. For a single NAND gate, let X_1 and X_2 be the signal probabilities of its two independent inputs. By assuming first that the NAND gate is fault-free, the output probability Z is given by $1 - X_1X_2$. If the gate has a probability ε of making an error, the output probability is then

$$\begin{aligned} Z = {} & P(\text{“1”}|\text{gate faulty})^{*}P(\text{gate faulty}) \\ & + P(\text{“1”}|\text{gate not faulty})^{*}P(\text{gate not faulty}). \end{aligned} \tag{2}$$

For gate errors of the von Neumann (inversion), stuck-at-0 and stuck-at-1 types, the output probabilities are, respectively,

$$Z_v = (1-\varepsilon)(1 - X_1X_2) + \varepsilon X_1X_2 = (1-\varepsilon) + (2\varepsilon - 1)X_1X_2, \tag{3}$$

$$Z_0 = (1 - X_1X_2)(1-\varepsilon), \tag{4}$$

$$Z_1 = 1 - (1-\varepsilon)X_1X_2. \tag{5}$$

In a PGM, a gate failure is actually modeled as an error at a gate's output, while an interconnect failure can be modeled as an input error of the gate into which this interconnect leads. Thus an input, or an interconnect, can be seen as an identity gate, which reproduces its input as output and is subject to the same error rate as other logic gates. Hence, the output probability of an unreliable interconnect for a von Neumann error is given by:

$$Z_v = X + \varepsilon(1 - 2X). \tag{6}$$

The PGMs of interconnect and several logic gates are given in Table 1. Note that in a nanoelectronic circuit, it may not be true that all gates and interconnects have the same error rate – the fault mechanisms and the rates of fault occurrence vary greatly depending on the actual implementation.

Please cite this article in press as: Han J et al. Reliability evaluation of logic circuits using probabilistic gate models. Microelectron Reliab (2010), doi:10.1016/j.microrel.2010.07.154

Table 1
Probabilistic gate models (PGMs) of Logic Gates.

	von-Neumann (Z_v)	Stuck-at-0 (Z_0)	Stuck-at-1 (Z_1)
NAND	$(1-\varepsilon)+(2\varepsilon-1)X_1X_2$	$(1-X_1X_2)(1-\varepsilon)$	$1-(1-\varepsilon)X_1X_2$
NOR	$1-X_2-X_1+X_1X_2(1-2\varepsilon)+\varepsilon(2X_1+2X_2-1)$	$1-X_2-X_1+X_1X_2+\varepsilon(X_1+X_2-X_1+X_2-1)$	$1-X_2-X_1+X_1X_2+\varepsilon(X_1+X_2-X_1X_2)$
Majority	$X_1X_2+X_1X_3+X_2+X_3-2X_1X_2X_3+\varepsilon(4X_1X_2X_3-2X_1X_2-2X_1X_3-2X_2X_3+1)$	$(1-\varepsilon)(X_1X_2+X_1X_3+X_2X_3-2X_1X_2X_3)$	$(1-\varepsilon)(X_1X_2+X_1X_3+X_2X_3-2X_1X_2X_3)+\varepsilon$
XOR	$X_2+X_1+2X_1X_2+\varepsilon(4X_1X_2-2X_2-2X_1+1)$	$X_2+X_1+2X_1X_2+\varepsilon(2X_1X_2-X_2-X_1)$	$X_2+X_1+2X_1X_2+\varepsilon(2X_1X_2-X_2-X_1+1)$
NOT	$1-X-\varepsilon+2\varepsilon X$	$1-X-\varepsilon+\varepsilon X$	$1-X+\varepsilon X$
Interconnect	$X+\varepsilon(1-2X)$	$X(1-\varepsilon)$	$X(1-\varepsilon)+\varepsilon$

2.2. A general procedure for the generation of PGMs

For a circuit that performs an arbitrary logic function, we can derive a probabilistic equation for this function, regardless of the specific structure of the circuit. A PGM of an arbitrary function can be constructed as follows.

As the symbolic logic (Boolean) operations map into the operations of a Boolean ring, the operations of binary functions can be mapped into a set of probability values in the real interval [0, 1] as follows [9,11]:

$$\overline{X} \to 1-X;$$
$$X_1 \wedge X_2 \to X_1X_2;$$
$$X_1 \vee X_2 \to X_1+X_2-X_1X_2.$$

These rules assign signal probabilities to the inputs and output of a gate or a functional unit, thus mapping the Boolean logic variables and operations into real-valued probabilities and arithmetic operations.

A general procedure to generate PGMs is as follows:

1. Construct a truth table that describes the Boolean function of the circuit (here, as a functional unit);
2. Use the Boolean ring mappings to obtain a sum of the minterms that produce a "1." The complement of this sum will produce a "0" when the unit is fault-free.
3. Consider the effect of the malfunction of the unit and, according to formula (1), sum over the conditional probabilities with and without function errors.

Let us take a 1-bit full adder as an example. The function for its sum output (shown in Table 2) can be described as:

$$S = (\overline{A} \wedge \overline{B} \wedge C_{in}) \vee (\overline{A} \wedge B \wedge \overline{C_{in}}) \vee (A \wedge \overline{B} \wedge \overline{C_{in}}) \vee (A \wedge B \wedge C_{in}).$$

According to the Boolean ring mappings, the sum of the minterms for output "1" and output "0" are

$$S = (1-A)(1-B)C_{in} + (1-A)B(1-C_{in}) + A(1-B)(1-C_{in}) + ABC_{in}, \quad (7)$$

and $1-S$ respectively.

Table 2
Truth table for the sum of an adder.

A	B	C_{in}	S
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

If the adder has a probability ε_c of making a von Neumann error, the output probability is given by:

$$\begin{aligned} S_v &= S(1-\varepsilon_c) + (1-S)\varepsilon_c \\ &= (1-\varepsilon_c)((1-A)(1-B)C_{in} + (1-A)B(1-C_{in}) \\ &\quad + A(1-B)(1-C_{in}) + ABC_{in}) + \varepsilon_c(1-((1-A)(1-B)C_{in} \\ &\quad + (1-A)B(1-C_{in}) + A(1-B)(1-C_{in}) + ABC_{in})). \end{aligned} \quad (8)$$

For stuck-at-0 and stuck-at-1 faults, the output probabilities are:

$$\begin{aligned} S_0 &= (1-\varepsilon_c)((1-A)(1-B)C_{in} + (1-A)B(1-C_{in}) \\ &\quad + A(1-B)(1-C_{in}) + ABC_{in}), \end{aligned} \quad (9)$$

$$\begin{aligned} S_1 &= (1-\varepsilon_c)((1-A)(1-B)C_{in} + (1-A)B(1-C_{in}) \\ &\quad + A(1-B)(1-C_{in}) + ABC_{in}) + \varepsilon_c. \end{aligned} \quad (10)$$

3. Reliability evaluation using probabilistic gate models (PGMS)

3.1. A simple algorithm

We first present a simple approach to reliability modeling. It is based on the iterative execution of PGMs guided by a specific circuit structure [27]. Consider a gate indexed i in a circuit. For von Neumann faults, the procedure to obtain the gate's PGM can be formulated as:

$$X_i = [p_i \quad 1-p_i] \cdot \begin{bmatrix} 1-\varepsilon \\ \varepsilon \end{bmatrix}, \quad (11)$$

where p_i is the sum of the minterms of inputs that produce a "1." For example, $p_i = 1 - X_{i-1}X_{i-2}$ for a NAND gate with inputs X_{i-1} and X_{i-2}. Here X_i is the output of gate i and thus an input to another gate in the circuit. Since every gate can be modeled by Eq. (11), an iterative execution of this procedure from a circuit's primary inputs to an output will produce an output probability.

A procedure for this simple algorithm is as follows:

1. Partition the circuit into m sub-circuits for each output, with m being the number of outputs of the circuit.
2. Assign initial signal probabilities to the primary inputs as well as error rates to the gates.
3. Proceeding from primary inputs to outputs, compute the signal probability of each output following the gates' PGMs.
4. Apply Eq. (1) to obtain the reliability of each output from the computed output probability.

In this method, a circuit is actually divided into small modules of single gates, and input/output signals are assumed to be statistically independent. Hence, an "overall" reliability for a circuit can be obtained by multiplying the individual reliabilities for each output. In a circuit that includes fanouts, however, signals are usually correlated rather than independent. Thus in the general case, this simple algorithm using PGMs will lead to approximate results.

Please cite this article in press as: Han J et al. Reliability evaluation of logic circuits using probabilistic gate models. Microelectron Reliab (2010), doi:10.1016/j.microrel.2010.07.154

3.2. An accurate algorithm

3.2.1. Handling disjoint reconvergent fanouts

The statistical dependence among signals comes from several sources. In a circuit without feedbacks, reconvergent fanouts are the only topological structure that induces signal dependence, provided that the inputs are independently distributed. Take the circuit in Fig. 1a as an example [28].

This circuit contains a fanout that originates from input B and reconverges at the output of gate G3. Because of the possible correlations caused by the fanout, the inputs of the gate G3 may not be statistically independent. However, if the signal probability of B is 0 or 1, i.e., B has a deterministic value ($B = 0$ or 1), the statistical dependence of the two inputs of gate G3 is actually eliminated. This is due to the removal of the probabilistic characteristics of the fanout. In effect, the circuit becomes one that does not contain the reconvergent fanout, as shown in Fig. 1b. The following lemma states when the signal dependence can be removed in a circuit containing reconvergent fanouts.

Lemma 1. *In a circuit without feedbacks, where all primary inputs are mutually independent, if the input to each reconvergent fanout is either "0" or "1," then all inputs to all gates in the circuit are mutually independent.*

Proof. Assume that an input to a reconvergent fanout is given by I and its branches are $X_1, X_2, \ldots, X_n$, where n is the number of fanout branches. We show that $X_1, X_2, \ldots, X_n$ are mutually independent by proving that $P(X_1, X_2, \ldots, X_n) = P(X_1)\,P(X_2)\ldots P(X_n)$ when $I = 0$ and $I = 1$. For $I = 0$, we have $P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = 1$ when $x_1 = x_2 = \cdots = x_n = 0$, and $P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = 0$ otherwise. Since $P(I) = P(X_1) = P(X_2) = \cdots = P(X_n)$, we have $P(X_1 = x_1)\,P(X_2 = x_2)\ldots P(X_n = x_n) = 1$ when $x_1 = x_2 = \cdots = x_n = 0$, and $P(X_1 = x_1)\,P(X_2 = x_2)\ldots P(X_n = x_n) = 0$ otherwise. Therefore, we have $P(X_1, X_2, \ldots, X_n) = P(X_1)\,P(X_2)\ldots P(X_n)$ for $I = 0$. For $I = 1$, the same conclusion can be obtained similarly. Hence, for each reconvergent fanout with input "0" or "1," its branch signals are mutually independent. Since reconvergent fanouts are the only source of signal dependence, all signals in this circuit are mutually independent. □

With the signal dependence in a circuit removed, it becomes possible to accurately evaluate the output probability of an arbitrary circuit. This is explained in the following theorem:

Theorem 1. *Given the input probability P_i of a reconvergent fanout F_i, the output probability of the circuit is given by $Z = Z_i^1 P_i + Z_i^0(1 - P_i)$, where Z_i^1 and Z_i^0 are the output probabilities when the input of the fanout F_i is set to "1" and "0," respectively.*

Proof. If the input to the fanout F_i is denoted as I_i, the output probability is then given by $Z = P(\text{output "1"}|I_i = 1) * P(I_i = 1) + P(\text{output "1"}|I_i = 0) * P(I_i = 0)$, that is $Z = Z_i^1 P_i + Z_i^0(1 - P_i)$. □

The single reconvergent fanout in Fig. 1a, for example, is handled by calculating the output probabilities of the two auxiliary networks in Fig. 1b – one with the input of the fanout set to "0" (Z^0) and the other with the input set to "1" (Z^1) – using the simple PGM algorithm. If the input of B has an actual probability P being a logical "1," the output probability is then given by $Z = Z^1 P + Z^0(1 - P)$. Multiple reconvergent fanouts can be handled in a similar way by considering each fanout individually, if the reconvergent fanouts are disjoint and independent of each other.

3.2.2. Handling dependent reconvergent fanouts

More complicated scenarios arise, however, when there are multiple reconvergent fanouts that are dependent in a circuit. The dependence among fanouts can be through the interwoven branches of inputs, as the one shown in Fig. 2a, or through a nested structure of hierarchical fanouts, as the one in Fig. 2b.

A general rule to handle multiple fanouts is to sequentially pick one fanout at a time and to treat them individually. For dependent fanouts positioned in parallel, this selection can, in principle, be in an arbitrary order. For the circuit in Fig. 2a, for instance, we first take fanout I_1. The output probability is then given by $J = J_1^1 I_1 + J_1^0(1 - I_1)$, where J_1^1 and J_1^0 are the output probabilities when I_1 is set to "1" and "0," respectively. Due to the presence of fanout I_2, J_1^1 and J_1^0 cannot be directly computed. However, further application of Theorem 1 to the fanout I_2, gives us $J_1^1 = J_{1,2}^{1,1} I_2 + J_{1,2}^{1,0}(1 - I_2)$ and $J_1^0 = J_{1,2}^{0,1} I_2 + J_{1,2}^{0,0}(1 - I_2)$, where $J_{1,2}^{1,1}$ denotes the output probability when $I_1 = 1$ and $I_2 = 1$, $J_{1,2}^{1,0}$ denotes the probability when $I_1 = 1$ and $I_2 = 0$, $J_{1,2}^{0,1}$ denotes the probability when $I_1 = 0$ and $I_2 = 1$ and $J_{1,2}^{0,0}$ denotes the probability when $I_1 = 0$ and $I_2 = 0$. Since I_2 is the last fanout in this circuit, all the elements in the above equations can be directly computed.

In circuits with nested fanout structures, each fanout is selected for evaluation in an order that proceeds from the primary inputs to the outputs of the circuit. For the circuit in Fig. 2b, by applying Theorem 1 to fanout I_2 first, we have $J = J_2^1 I_2 + J_2^0(1 - I_2)$. Proceeding to fanout P_4 gives us $J_2^1 = J_{2,4}^{1,1} P_{4,I_2=1} + J_{2,4}^{1,0}(1 - P_{4,I_2=1})$, and $J_2^0 = J_{2,4}^{0,1} P_{4,I_2=0} + J_{2,4}^{0,0}(1 - P_{4,I_2=0})$, where $J_{2,4}^{1,1}$ denotes the output probability when $I_2 = 1$ and $P_4 = 1$, $J_{2,4}^{1,0}$ denotes the probability when $I_2 = 1$ and $P_4 = 0$, $J_{2,4}^{0,1}$ denotes the probability when $I_2 = 0$ and $P_4 = 1$ and $J_{2,4}^{0,0}$ denotes the probability when $I_2 = 0$ and $P_4 = 0$; $P_{4,I_2} = 1$ is the conditional probability of P_4 when $I_2 = 1$, and $P_{4,I_2} = 0$ is the conditional probability of P_4 when $I_2 = 0$.

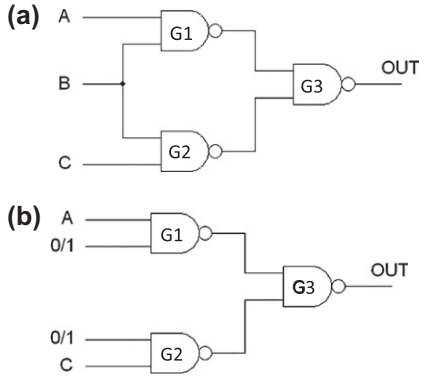

Fig. 1. (a) A simple circuit containing a reconvergent fanout. (b) Equivalent circuit without the fanout.

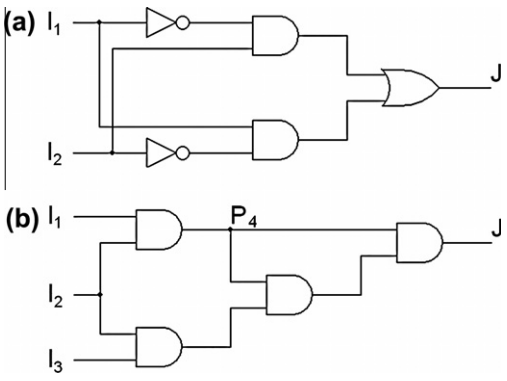

Fig. 2. Circuits that contain dependent, reconvergent fanouts: (a) parallel structure and (b) nested structure.

Please cite this article in press as: Han J et al. Reliability evaluation of logic circuits using probabilistic gate models. Microelectron Reliab (2010), doi:10.1016/j.microrel.2010.07.154

A generalization of this analysis is stated as follows:

Corollary 1. *In a circuit with n dependent reconvergent fanouts, its output probability is given by* $Z = \sum_{j=0}^{2^n-1} Z_j P_j$, *where* $P_j = P(\text{inputs} = j)$, *i.e., the probability that the input vector to the fanouts has the jth value, and* $Z_j = P(\text{output} = 1 | \text{inputs} = j)$, *i.e., the conditional probability that the output is "1," given that the input vector is j.*

The proof is straightforward from Theorem 1 and the above analysis. For the two parallel fanouts in Fig. 2a, for example, by Corollary 1 we obtain an output probability of $J = J_{1,2}^{1,1} I_1 I_2 + J_{1,2}^{1,0} I_1 (1 - I_2) + J_{1,2}^{0,1} (1 - I_1) I_2 + J_{1,2}^{0,0} (1 - I_1)(1 - I_2)$. In the nested case of Fig. 2b, a similar expression can be obtained, however, the input probability of the intermediate fanout P_4 is in the form of the conditional probability that is dependent on and calculated from the first fanout I_2.

3.2.3. A general procedure and an example

In general, by each application of Theorem 1 over a fanout, the original circuit is effectively reduced to two auxiliary circuits, each of which has one less fanout. An iterative execution of this process over all dependent fanouts produces a total of 2^{N_f} auxiliary circuits, where N_f is the total number of reconvergent fanouts that are dependent in the circuit. As a result, the conditional input probability of each reconvergent fanout $P(X_{N_i} \mid X_{N_i-1}, X_{N_i-2}, \ldots, X_1)$ for $N_i \leqslant N_f$, given the inputs of its previous dependent fanouts $(X_{N_i-1}, X_{N_i-2}, \ldots, X_1)$ set to "0" and "1," and the output probability of the last reconvergent fanout for each combination of the inputs of all dependent fanouts $P(J_{N_f} \mid X_{N_f}, X_{N_f-1}, \ldots, X_1)$, are indispensable. The procedure therefore involves the computation of the conditional input probabilities of all dependent fanouts and the conditional output probabilities of the last fanout, both of which now can be obtained using the simple PGM algorithm.

For an arbitrary circuit with m outputs, a general procedure for the accurate algorithm to obtain the output reliabilities is as follows:

1. Partition the circuit into m sub-circuits for each output.
2. Separate the fanouts into two groups: single reconvergent fanouts that are disjoint to others and multiple reconvergent fanouts that are dependent on each other.
3. Compute the input probability and then the output probability of each disjoint reconvergent fanout.
4. For each set of dependent reconvergent fanouts, compute an output probability for each combination of the inputs of all fanouts, which have been set to "0" and "1," using the simple algorithm.
5. For each set of dependent reconvergent fanouts, proceeding from inputs to the output, compute a conditional probability for the input of each fanout, given the inputs of its previous fanouts set to "0" and "1," using the simple algorithm.
6. For each set of dependent reconvergent fanouts, proceeding from the output to inputs, compute the output probability by a recursive procedure as specified in Theorem 1, using the initial output probabilities obtained in Step 4 and the conditional input probabilities obtained in Step 5.
7. Compute the overall probability of each output of the circuit, using the simple algorithm.
8. Apply Eq. (1) to obtain the reliability of each output from the computed output probability.

An algorithmic flowchart of the accurate PGM approach is shown in Fig. 3. We further illustrate it using a simple ISCAS85 benchmark circuit C17 as follows. As shown in Fig. 4, C17 has two primary outputs, so it is first partitioned into two sub-circuits for each output. Tracing back from each primary output, the reconvergent fanouts F1 and F2 are identified with respect to outputs N22 and N23 respectively. Then, by decomposing each fanout into sub-circuits with inputs "0" and "1," we compute the conditional output and input probabilities of each fanout. With these conditional probabilities, the probability and reliability of N22 and N23 can be obtained by applying Theorem 1 and the simple PGM algorithm.

Given the netlist of a circuit, which specifies the circuit's characteristics such as the numbers of inputs and outputs, the number of gates and their connectivity, the fanout identification and decomposition inevitably lead to revised netlists of the circuit and its sub-circuits. Once generated, however, the new netlists reflect and trace the signal dependencies introduced by reconvergent fanouts. The evaluation process becomes then straightforward based on these new netlists, and can readily be implemented into a software package.

3.2.4. Handling correlated inputs

So far we have assumed that the inputs of circuits are independent. However, correlations in inputs can be modeled in a similar way as are the correlations due to reconvergent fanouts. When correlated, the inputs share joint distributions and Corollary 1 can be adapted to apply to correlated inputs.

Corollary 2. *In a circuit with n inputs that may be correlated, its output probability is given by* $Z = \sum_{j=0}^{2^n-1} Z_k P_k$, *where* $P_k = P(\text{inputs} = k)$, *i.e., the probability that the input vector to the circuit has the kth value, and* $Z_k = P(\text{output} = 1 | \text{inputs} = k)$, *i.e., the conditional probability that the output is "1," given that the input vector is k.*

When the correlations in inputs are accounted for, the circuit is left with correlations that are only due to reconvergent fanouts and can thus be modeled by the procedure outlined in Section 3.2.3.

3.3. Complexity issues

The PGM algorithms compute circuit reliability for each input combination and each output. For a circuit with n inputs and m outputs, therefore, time complexities of the PGM algorithms are polynomial in the number of outputs and exponential in the number of inputs, that is a complexity of $O(m \cdot 2^n)$.

For each output and input pattern, the time complexity can be evaluated by assessing the number of computational steps involved in each algorithm. For the simple algorithm, a circuit's reliability is calculated by an exhaustive implementation of the PGMs over all gates in the circuit. Since a logic gate typically has a few inputs and one output, we can consider, for the purpose of complexity evaluations, that the time complexity of each gate model is similar and only requires a limited number of computation steps. Hence, the time complexity of the simple PGM algorithm increases linearly with the number of gates in a circuit, i.e. a polynomial complexity of $O(N)$, where N is the total number of gates in a circuit.

For the accurate algorithm, since disjoint, reconvergent fanouts are considered individually, the computation steps needed increase in a polynomial order with the number of disjoint fanouts. For dependent, reconvergent fanouts, there are three major steps: the computation of output probabilities for all input combinations (Step 4 in the algorithm procedure), the computation of conditional probabilities of the inputs of intermediate fanouts (Step 5) and the recursive computation of the final output probability (Step 6). For N_f dependent, reconvergent fanouts in the worst case, there are a total of 2^{N_f} output probabilities that need to be computed in Step 4. Since the complexity involved in computing one output probability is $O(N)$ by using the simple algorithm, the complexity of Step 4 is:

$$O(N2^{N_f}). \tag{12}$$

Please cite this article in press as: Han J et al. Reliability evaluation of logic circuits using probabilistic gate models. Microelectron Reliab (2010), doi:10.1016/j.microrel.2010.07.154

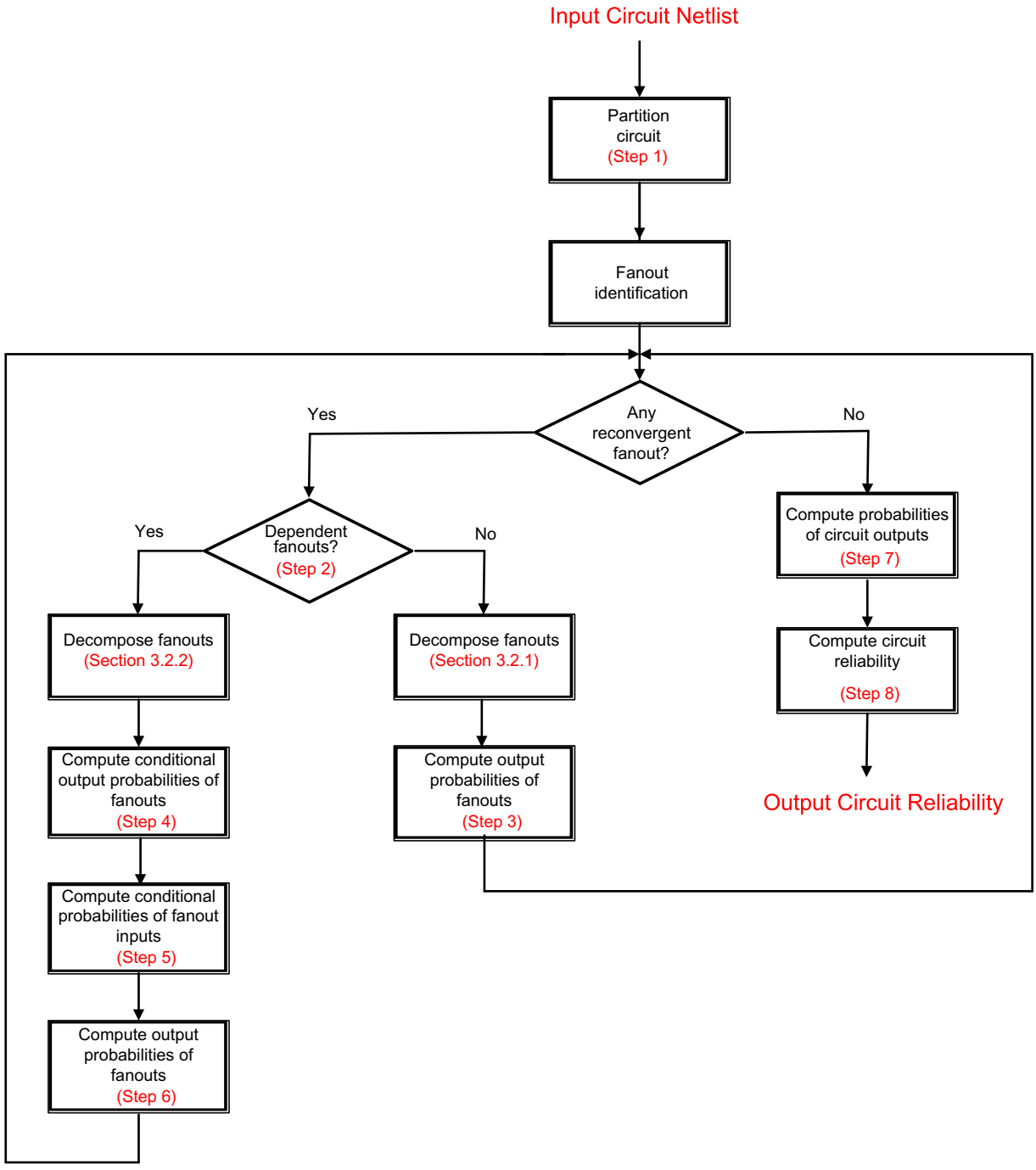

Fig. 3. An algorithmic flowchart of the accurate PGM approach.

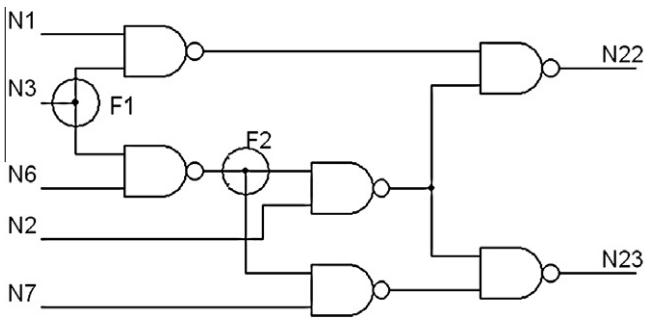

Fig. 4. Schematic of C17.

For each fanout in Step 5, there are 2^{N_i} $(N_i < N_f)$ conditional probabilities due to its previous N_i reconvergent fanouts. Therefore, the number of computation steps is given by:

$$\sum_{N_i=1}^{N_f-1}(C_{N_i}2^{N_i}), \tag{13}$$

where C_{N_i} is the number of computations needed for computing the conditional probability for fanout N_i. Since C_{N_i} is not larger than $O(N)$, (13) is bounded by

$$O(N\sum_{N_i=1}^{N_f-1}2^{N_i}). \tag{14}$$

Please cite this article in press as: Han J et al. Reliability evaluation of logic circuits using probabilistic gate models. Microelectron Reliab (2010), doi:10.1016/j.microrel.2010.07.154

The last step involves the computation of the circuit output probability using Theorem 1. For each intermediate fanout, the number of computation steps, as specified in Theorem 1, depends on the number of reconvergent fanouts that precede it, given by $O(2^{N_i})$. So the total number of computations is given by:

$$O(\sum_{N_i=1}^{N_f} 2^{N_i}). \quad (15)$$

Suppressing the components of lower powers in (12), (14), and (15) gives us a complexity of

$$O(N2^{N_f}), \quad (16)$$

i.e., in the worst case when all of the reconvergent fanouts in a circuit are dependent, the time complexity of the accurate PGM algorithm is polynomial in the number of gates and exponential in the number of reconvergent fanouts.

3.4. Simulation results

To demonstrate the accuracy and efficiency of the proposed algorithms, we first show simulation results on a number of small circuits. The overall reliabilities obtained using the simple and accurate PGM algorithms, as well as the circuits' characteristics, are listed in Table 3 for a gate error rate of 0.05 (von Neumann errors). The results from the PTM approach are also shown for comparison. The simulations were run on a 2.66-GHz Pentium microprocessor with 2 GB memory. It can be seen that the accurate PGM algorithm provides the same results as those obtained by the PTM approach, but generally requires a smaller runtime. The simple PGM algorithm requires an even shorter runtime, and provides highly accurate reliability estimates for circuits of such sizes. In the special cases of majority, full adders (Majority and XOR/NAND) and decoder2, where the reconvergent fanouts originate at the primary inputs and the primary inputs are assumed to be fault-free, the simple algorithm produces the same reliability with the same runtime as those of the accurate algorithm.

We further present our investigation on the LGSynth91 benchmark circuits. The results are shown in Table 4. The inputs were assumed to be uniformly distributed and up to 1000 random samples were used for each circuit. As revealed in the table, the simple PGM algorithm that does not consider signal dependencies introduced by fanouts provides very good approximations of the exact reliabilities obtained by using the accurate PGM algorithm that does model fanout behavior. For some of the circuits, namely, the pcle, decod, parity and pm1, the simple algorithm provides the same evaluation results as those obtained by the accurate algorithm. This is because (1) the parity has a tree structure and thus does not contain any reconvergent fanout and (2) in the other circuits, all reconvergent fanouts originate at the primary inputs of the circuit, except for the pcle that has parallel recovergent fanouts. In case (2), since perfectly reliable inputs were considered, the reconvergent fanouts that started from the primary inputs did not cause any signal correlation (per Lemma 1) and were therefore ignored in our simulations. As a result, the simple and accurate PGM approaches produced the same reliability and runtime values

Table 3
Comparisons of evaluation results using the simple PGM, accurate PGM and PTM algorithms ($\varepsilon = 0.05$).

Circuit	Characteristics			Simple PGM algorithm			Accurate PGM algorithm		The PTM approach	
	Gates	Inputs	Outputs	Reliability	Runtime	Relative error	Reliability	Runtime	Reliability	Runtime
C17	6	5	2	0.7621	0.004s	0.51%	0.7582	0.013s	0.7582	0.02s
Majority	10	5	1	0.8623	0.02s	0%	0.8623	0.02s	0.8623	0.06s
Full adder (Majority)	8	3	2	0.7904	0.01s	0%	0.7904	0.01s	0.7904	0.24s
Full adder (XOR/NAND)	6	3	2	0.7879	0.01s	0%	0.7879	0.01s	0.7879	0.01s
Full adder (NAND)	12	3	2	0.5933	0.001s	1.26%	0.6009	0.01s	0.6009	0.02s
Comparator	4	2	3	0.7292	0.0002s	0.96%	0.7363	0.005s	0.7363	0.01s
Decoder2	6	2	4	0.7397	0.009s	0%	0.7397	0.009s	0.7397	0.013s
MUX4	7	6	1	0.8222	0.002s	0.01%	0.8221	0.01s	0.8221	0.44s

Table 4
Evaluation results for the LGSynth91 benchmark circuits using the PGM algorithms ($\varepsilon = 0.05$).

Circuit	Characteristics			Accurate PGM algorithm		Simple PGM algorithm		
	Gates	Inputs	Outputs	Reliability	Runtime	Reliability	Runtime	Relative error
cu	43	14	11	0.3865	0.100625s	0.3812	0.027655s	1.37%
z4 ml	45	7	4	0.2546	0.051828s	0.2522	0.003893s	0.943%
pcle	61	19	9	0.2342	0.068082s	0.2342	0.033649s	0%
decod	22	5	16	0.3426	0.000996s	0.3426	0.000996s	0%
parity	15	16	1	0.6029	0.012509s	0.6029	0.012509s	0%
pm1	41	16	13	0.3886	0.026850s	0.3886	0.026850s	0%
x2	38	10	7	0.3822	0.218194s	0.3802	0.022748s	0.523%
mux	50	21	1	0.7920	0.098271s	0.7961	0.028193s	0.518%

Table 5
Evaluation results for the 74X-Series circuits using the PGM algorithms ($\varepsilon = 0.05$).

Circuits	Characteristics			Average reliability	
	Gates	Inputs	Outputs	Simple PGM algorithm	Accurate PGM algorithm
74182	19	9	4	0.867	0.867
74283	36	9	5	0.784	0.757
74181	61	14	8	0.725	0.697

Please cite this article in press as: Han J et al. Reliability evaluation of logic circuits using probabilistic gate models. Microelectron Reliab (2010), doi:10.1016/j.microrel.2010.07.154

for these circuits. Although not shown here, highly accurate results were also obtained from our evaluations of these circuits using imperfect inputs.

In general, the relative errors introduced by the simple algorithm are less than 1% with an exception of 1.37%. While offering accurate evaluation results, the accurate algorithm requires a significantly increased runtime of up to 13-times of the time required by the simple algorithm. Plus, the runtime reported in Table 4 does not include the much longer time needed in identifying fanout paths and decomposing the circuit according to the sometimes complicated circuit topologies. For the LGSynth91 circuits, as has been seen, the simple algorithm provides highly accurate evaluation results, especially for those without or with only a few reconvergent fanouts.

The PGM algorithms were also used to evaluate benchmark circuits from the 74X-Series including a 4-bit carry-look ahead generator (74182), a 4-bit adder (74283) and a 4-bit ALU (74181). The average reliabilities obtained for these circuits are shown in Table 5. It can be seen that the accurate and simple algorithms produce the same result for the 74182 circuit (in which fanouts originate at primary inputs), while for the other two circuits, the simple PGM only gives an approximation of the reliability obtained by the accurate algorithm.

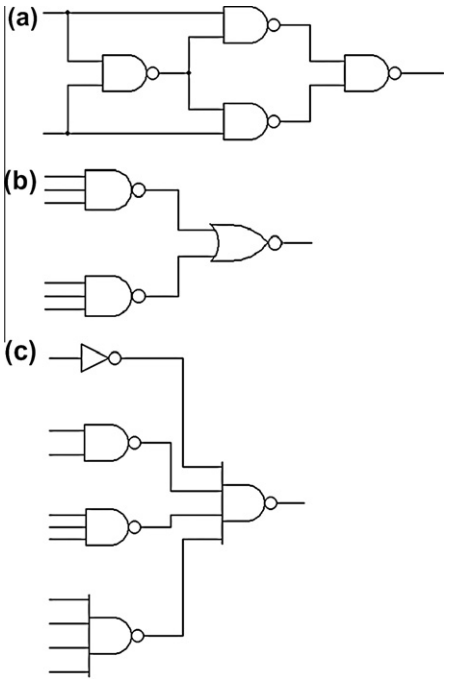

Fig. 5. Common logic modules found in circuits include: (a) 2-input XOR, (b) 6-input M1, and (c) carry generator for third bit of CLA [29].

4. A modular approach

In this section, we present a modular approach as a tradeoff between the simplicity of the simple PGM algorithm and the accuracy of the accurate PGM algorithm. As noted in [29], many large circuits such as the ISCAS85 benchmarks contain a limited number of simple logic components that are used repeatedly throughout a design. Some benchmarks frequently utilize functional modules such as those shown in Fig. 5. Generally, an accurate evaluation of the reliability of such large circuits is time-intensive as complexity grows with the number of gates and signal dependencies. However, due to the presence of common logic components used throughout a design, the analysis of reliability can be simplified.

As in most practical applications, the modules contained in a circuit are known *a priori*, reliability equations can be constructed for such modules by using the accurate PGM algorithm. With these equations, a reliability estimate for an output, denoted by *X*, can be obtained by determining the logic modules that lie along the path from the primary inputs to *X*. The equations for these modules can then be applied in a topological order without regard for signal dependencies between different modules. Finally, the output of the last module is used to estimate the reliability of *X* as well as the reliability of the circuit path from the primary inputs to *X*. This method, hence, does not only give us estimates for reliability, but also finds the most vulnerable outputs and paths in a circuit, which is often useful when evaluating the necessity to apply fault tolerant techniques to improve reliability. For circuits that contain large modules, this method can be applied recursively by first decomposing the large modules into smaller components which themselves may be composed of even smaller blocks of logic. Thus, the hierarchical structure of a circuit can be exploited to efficiently apply this modular approach.

To illustrate this technique, we consider the 74283 circuit, a 4-bit adder with 9 inputs, 5 outputs and 36 gates. In our evaluation, we set $\varepsilon = 0.05$ and assume that all 2^9 input combinations are equally probable. We note that this circuit is composed of two common logic components, M1 (Fig. 5b) and a carry-lookahead generator (CLA). The CLA itself is a large module, so it is broken down further into four smaller functional blocks, CLA1–CLA4 (Fig. 5c). In this way, the circuit outputs can be obtained by modeling the combinations of the M1 and CLA1–CLA4 modules. Our evaluation results indicate that the modular approach gives an average relative error of 0.4% compared to the accurate approach, while the runtime is reduced by 97.3% for the 74283. With this reduced complexity, this approach can be used to estimate the reliability of large circuits, as well as to identify the most vulnerable path in a circuit.

The modular approach has been applied to the evaluation of the larger ISCAS85 benchmark circuits. The average reliabilities of the circuits are shown in Table 6 and are compared to those obtained by Monte Carlo (MC) simulations. MC simulations have been used as the standard method and known to provide the best estimate of circuit reliability when accurate analytical approaches are not applicable. As revealed in Table 6, the modular PGM approach provides fairly accurate reliability estimates while it only requires a runtime that is several orders of magnitude (10^3–10^5) lower than that of the MC simulations. In the MC simulation, a total of one million random patterns were generated for each circuit in order to guarantee a relatively stable evaluation result. This stability can be measured by the standard deviation of the data. In our simulation, the standard deviation is in the order of 10^{-4} for one million runs, which indicates an error rate of less than 0.1% in the obtained reliability value. The runtime can be reduced by using a smaller number of runs, however, this would introduce more fluctuations into the evaluation results obtained by the MC simulations. For the modular approach, the breaking down in complexity due to the decomposition of the circuits into smaller modules significantly reduces the runtime, as well as the evaluation effort, required by the analyses of large circuits. These results demonstrate the accuracy and efficiency of the modular approach. This technique is especially useful for circuits that have a regular structure of common modules, such as the C6288 composed of half and full adders in a grid structure. It was also shown in our results that the modular approach correctly predicted the least reliable outputs of the circuits.

Since reliability is becoming a major design metric in nanoscale VLSI circuits and systems, the modular approach is especially useful and is ready to be incorporated into the early design flow. In

Please cite this article in press as: Han J et al. Reliability evaluation of logic circuits using probabilistic gate models. Microelectron Reliab (2010), doi:10.1016/j.microrel.2010.07.154

Table 6
Evaluation Results for the ISCAS85 benchmark circuits using the modular approach and Monte Carlo (MC) simulation ($\varepsilon = 0.05$ and 1 million runs for the MC simulation).

Circuits	Characteristics			Modular PGM approach			Monte Carlo simulation	
	Gates	Inputs	Outputs	Average reliability	Runtime	Relative error	Average reliability	Runtime
C432	160	36	7	0.7434	0.25s	9.21%	0.6807	49m
C499	202	41	32	0.8764	2.15s	0.11%	0.8754	52.6m
C1355	546	41	32	0.8078	2.98s	4.2%	0.7752	181.7m
C2670	1193	157	64	0.8011	4.26s	0.43%	0.8046	461.7m
C6288	2416	32	32	0.5626	2.06s	4.57%	0.5380	693.3m

current digital design, a critical tool for managing complexity is through the use of hierarchy – a large system is hierarchically partitioned into less complex sub-systems, a process that can be repeated to obtain modular functional blocks of processors, circuits, gates and transistors. At each level of the abstraction, hence, reliability analysis can readily be implemented following the procedure of the modular PGM approach. In addition to the hierarchical abstraction, modular regularity has also been explored for further reducing the complexity of a design, which makes the application of the modular approach even more efficient and effective.

5. Conclusions and discussion

As CMOS approaches its scaling limits and new nanoelectronics emerge, reliability becomes a major concern. To meet the need for reliability evaluation techniques, we present the notion of probabilistic gate models (PGMs) to obtain computational algorithms for evaluating circuit reliability. The approximate PGM algorithm provides an efficient way to estimate the reliability of logic circuits. It is suitable for the reliability evaluation of small circuits as well as large circuits where there are no or few reconvergent fanouts. The accurate PGM algorithm is able to accurately evaluate circuit reliability, while in the worst case it requires a time complexity that is exponential in the number of reconvergent fanouts.

Since many large circuits are composed of a limited number of sub-modules, a hierarchical, modular approach is proposed for practical applications. Large modules can be decomposed into their smaller components and then reliability estimates can be obtained by recursively applying the PGM algorithm to this hierarchy of modules. Our results indicate that this approach produces reasonably accurate estimates for reliability with a significantly reduced runtime. Thus, the modular approach presents a tradeoff between accuracy and complexity in reliability evaluation.

Although not discussed in detail, the modular PGM approach can also be used for the reliability evaluation of sequential circuits. By using a so-called time-frame expansion technique [30], a sequential circuit is unrolled into a series of identical combinational modules connected in the spatial domain. The number of time-frames is determined by the specific clock cycles that are of one's interest. This is a particularly efficient procedure for the modeling of commonly used single-clock synchronous sequential circuits. Given the accuracy and scalability of the modular approach, the reliability of such a sequential circuit for an arbitrary time-frame can be obtained by considering the circuit as a number of modules connected in series. Hence, the proposed modular approach is potentially useful in the reliability analyses of both combinational and sequential circuits, as well as design-for-reliability applications where it is important to identify critical paths in a circuit.

References

[1] Borkar S. Designing reliable systems from unreliable components: the challenges of transistor variability and degradation. IEEE Micro 2005;25(6):10–6.
[2] Wong BP, Mittal A, Cao Y, Starr G. Nano-CMOS circuit and physical design. Hoboken: John Wiley & Sons; 2005.
[3] Constantinescu C. Trends and challenges in VLSI circuit reliability. IEEE Micro 2003:14–9.
[4] International Technology Roadmap for Semiconductors; 2009. <http://public.itrs.net/>.
[5] Shanbhag NR, Mitra S, de Veciana G, Orshansky M, Marculescu R, Roychowdhury J, Jones D, Rabaey JM. The search for alternative computational paradigms. The special issue on "System IC Design Challenges beyond 32 nm". IEEE Des Test Comput 2008;25(4).
[6] von Neumann J. Probabilistic logics and the synthesis of reliable organisms from unreliable components. In: Shannon CE, McCarthy J, editors. Automata studies. Princeton (NJ): Princeton University Press; 1956. p. 43–98.
[7] Siewiorek DP, Swarz RS. Reliable computer systems: design and evaluation. Natick (MA, USA): AK Peters; 1998.
[8] Shukla SK, Bahar RI, editors. Nano, quantum and molecular computing: implications to high level design and validation. Boston: Kluwer Academic Publishers; 2004.
[9] Parker KP, McCluskey EJ. Probabilistic treatment of general combinational networks. IEEE Trans Comput 1975;C(24):668–70.
[10] Bass SC, Grundmann JW. Expected value analysis of combinational logic networks. IEEE Trans Circuits Syst 1981;28(5).
[11] Bahar RI, Chen J, Mundy J. A probabilistic-based design methodology for nanoscale computation. In: Proceedings of international conference on computer aided design, San Jose, CA, November 2003. p. 480–6.
[12] Bhaduri D, Shukla S. Nanoprism: a tool for evaluating granularity versus reliability tradeoffs in nano architectures. In: ACM GLSVLSI, Boston, MA, April 2004.
[13] Krishnaswamy S, Viamontes GF, Markov IL, Hayes JP. Probabilistic transfer matrices in symbolic reliability analysis of logic circuits. ACM Trans Des Autom Electron Syst 2008;13(1#8).
[14] Rejimon T, Lingasubramanian K, Bhanja S. Probabilistic error modeling for nano-domain logic circuits. IEEE Trans VLSI 2009;17(1):55–65.
[15] Choudhury MR, Mohanram K. Reliability analysis of logic circuits. IEEE Trans CAD 2009;28(3):392–405.
[16] Abdollahi A. Probabilistic decision diagrams for exact probabilistic analysis. In: Proc of IEEE/ACM intl conference on computer aided design; 2007. p. 266–72.
[17] Mohyuddin N, Pakbaznia E, Pedram M. Probabilistic error propagation in logic circuits using the Boolean difference calculus. In: IEEE intl conf on comp des, Lake Tahoe, CA, USA; 2008. p. 7–13.
[18] Franco DT, Vasconcelos MC, Naviner L, Naviner J-F. Signal probability for reliability evaluation of logic circuits. Microelectron Reliab 2008;48(8-9):1586–91.
[19] Sivaswamy S, Bazargan K, Riedel M. Estimation and optimization of reliability of noisy digital circuits. In: International symposium on quality electronic design, San Jose, CA, USA; 2009. p. 213–9.
[20] Seyyed Mahdavi SJ, Mohammadi K. SCRAP: sequential circuits reliability analysis program. Microelectron Reliab 2009;49(8):924–33.
[21] Han J, Jonker P. A system architecture solution for unreliable nanoelectronic devices. IEEE Trans Nanotechnol 2002;1(4):201–8.
[22] Qi Y, Gao JB, Fortes JAB. Markov chains and probabilistic computation – a general framework for multiplexed nanoelectronic systems. IEEE Trans Nanotechnol 2005;4(2):194–205.
[23] Roy S, Beiu V. Majority multiplexing-economical redundant fault-tolerant designs for nanoarchitectures. IEEE Trans Nanotechnol 2005;4(4):441–51.
[24] Han J, Gao J, Qi Y, Jonker PP, Fortes JAB. Toward hardware-redundant, fault-tolerant logic for nanoelectronics. IEEE Des Test Comput 2005;22(4):328–39.
[25] Anghel L, Nicolaidis M. Defects tolerant logic gates for unreliable future nanotechnologies. In: Sandoval F et al., editors. Lecture notes in computer science. IWANN 2007 4507; 2007. p. 422–9.
[26] El-Maleh AH, Al-Hashimi BM, Melouki A, Khan F. Defect-tolerant N^2-transistor structure for reliable nanoelectronic designs. IET Comput Digit Tech 2009;3(6):570–80.
[27] Han J, Taylor ER, Gao J, Fortes JAB. Faults, error bounds and reliability of nanoelectronic circuits. In: Proc. IEEE ASAP 2005; 2005. p. 247–53.
[28] Taylor ER, Han J, Fortes JAB. Towards accurate and efficient reliability modeling of nanoelectronic circuits. In: Proc IEEE-NANO 2006. IEEE conference on nanotechnology.
[29] Hansen MC, Yalcin H, Hayes JP. Unveiling the ISCAS-85 benchmarks: a case study in reverse engineering. IEEE Des Test Comput 1999;16(3):72–80.
[30] Bushnell ML, Agrawal VD. Essentials of electronic testing. Boston: Springer; 2005.

Please cite this article in press as: Han J et al. Reliability evaluation of logic circuits using probabilistic gate models. Microelectron Reliab (2010), doi:10.1016/j.microrel.2010.07.154