Research Article

Programmed Tool for Quantifying Reliability and Its Application in Designing Circuit Systems

N. S. S. Singh

Fundamental and Applied Sciences Department, Universiti Teknologi PETRONAS, Bandar Seri Iskandar, 31750 Tronoh, Perak, Malaysia

Correspondence should be addressed to N. S. S. Singh; narinderjit@petronas.com.my

Received 19 January 2014; Revised 8 April 2014; Accepted 25 April 2014; Published 18 May 2014

Academic Editor: Muhammad Taher Abuelma'atti

Copyright © 2014 N. S. S. Singh. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As CMOS technology scales down to nanotechnologies, reliability continues to be a decisive subject in the design entry of nanotechnology-based circuit systems. As a result, several computational methodologies have been proposed to evaluate reliability of those circuit systems. However, the process of computing reliability has become very time consuming and troublesome as the computational complexity grows exponentially with the dimension of circuit systems. Therefore, being able to speed up the task of reliability analysis is fast becoming necessary in designing modern logic integrated circuits. For this purpose, the paper firstly looks into developing a MATLAB-based automated reliability tool by incorporating the generalized form of the existing computational approaches that can be found in the current literature. Secondly, a comparative study involving those existing computational approaches is carried out on a set of standard benchmark test circuits. Finally, the paper continues to find the exact error bound for individual faulty gates as it plays a significant role in the reliability of circuit systems.

1. Introduction

As CMOS technology scales down to nanotechnologies, inconsistency in the performance of transistors will keep on escalating, making the transistors less and less reliable. The variation of inconsistencies will subsequently affect the performance and degrade the reliability of circuit systems made up of these transistors [1]. Inconsistencies in the performance of transistors are due to random dopant fluctuation or manufacturing imprecision in the CMOS fabrication process. Due to this physical level of characteristics, probabilistic behaviors will continue to develop in the system which naturally affects the evolving technologies to have significant limitations for reliable operation.

Reliability has, therefore, become a major concern and thus present new challenges to the design of circuit systems. Various probabilistic design methodologies have been implemented to assemble reliable circuit systems out of unreliable devices [2, 3]. To meet this increasing demand on having reliable design, several computational approaches have been also proposed for the reliability evaluation of

nanotechnology-based circuits [4–13]. They are probabilistic gate model (PGM) [4–6], Boolean difference-based error calculator (BDEC) [7], probabilistic transfer matrix (PTM) [8–11], and Bayesian network (BN) [12, 13]. As computational complexity increases exponentially with circuit dimension, the whole reliability evaluation process becomes very tedious and time consuming. Therefore, for speedy reliability computation, this paper firstly looks into the development of an automated reliability evaluation tool based on the generalization of PGM, BDEC, PTM, and BN models. For demonstration purpose, in this paper, C17 has been chosen as the standard benchmark test circuit for its reliability computation based on PTM approach. For reliability computation, input to the automated MATLAB-based tool is a description of the desired circuit in its Netlist form that is comprised of gate profile matrix (GPM), adjacency computation matrix (ACM), and grid layout matrix (GLM). GPM indicates the types of faulty gates that exist in the desired circuit layout whereby ACM and GLM indicate the interconnection and layout permutation between these faulty gates, respectively.

TABLE 1: Truth table of a NOR gate.

Y_1	Y_2	Z
0	0	1
0	1	0
1	0	0
1	1	0

Secondly, the paper continues exploring a comparative study on the reliability computation time and storage complexity for a set of standard benchmark test circuits using PGM, BDEC, PTM, and BN models.

Finally, reliability of a desired circuit system is not only affected by its faulty gates, but it also depends on the size of error, P , in those faulty gates. In order for the system to be reliable, the size of error in its faulty gates has to be smaller than a threshold, P^* . Error thresholds for faulty gates are increasingly important subject to be considered in the fault tolerance computation as it gives the exact bounds above which no reliable computation is possible. For this purpose, we have shown here that apart from computing reliability of the desired circuit system, the automated tool is also capable of computing the fundamental error threshold for individual faulty gates using PTM approach.

2. Reliability Computational Approaches

In this section, reliability function for individual faulty gates and circuit reliability evaluation models, namely PGM, BDEC, PTM, and BN, are explained which include some of the basic definitions and terms in brief.

2.1. Reliability Function for Individual Faulty Gates. The simple Von Neumann model made an assumption that any faulty gate flips out of its output with a probability of gate error, $P \leq 0.5$, while the input and output lines function reliably [4]. For example, Table 1 shows the truth table of a NOR gate.

Y_1 , Y_2 , and Z denote the input probabilities and output probability of signal value being “1,” respectively. For a fault-free NOR gate, (gate error probability, $P = 0$), the probability of its output Z being “1” is $(1 - Y_1)(1 - Y_2)$. This can be obtained from the minterms that produce $Z = 1$. If the gate has the probability of making an error, P , then the probability of its output Z being “1” is shown in (1) [4, 5]. Using this technique, we can construct reliability function, Z for any logic gates:

$$\begin{aligned} Z &= (1 - P) [(1 - Y_1)(1 - Y_2)] + P [1 - (1 - Y_1)(1 - Y_2)] \\ &= 1 - Y_1 - Y_2 + Y_1 Y_2 (1 - 2P) + P (2Y_1 + 2Y_2 - 1). \end{aligned} \quad (1)$$

2.2. Probabilistic Gate Model (PGM). PGM is a reliability evaluation model for designing circuit systems. For computing reliability of the desired circuit system, PGM executes reliability function of each faulty gate right from the input to the output signals of the system [4–6]. In the desired

circuit layout, for a faulty gate indexed i , the gate’s PGM is represented by (2)

$$\text{PGM}_i = [p_i \ 1 - p_i] \begin{bmatrix} 1 - P \\ P \end{bmatrix}, \quad (2)$$

where p_i = sum of the input signals that produces a signal value being “1” and P = internal gate error probability.

In PGM, primary output signals are assumed to be statistically independent and uncorrelated [4–6]. Therefore, reliability of the circuit design can be achieved by multiplying all the individual output reliabilities as briefly discussed in [4–6].

2.3. Boolean Difference-Based Error Calculator (BDEC). Apart from PGM, BDEC is also known as a very powerful probabilistic model for circuit reliability evaluation. For computing reliability of the desired circuit system, BDEC propagates errors right from the input to the output signals in the presence errors on signal lines, e , and internal faulty gates, P [7]. For a two-input faulty gate in the desired circuit layout, the general equation of its output error probability, Z , based on BDEC model is represented by (3)

$$\begin{aligned} Z &= P + (1 - 2P) \left[e_1 (1 - e_2) \Pr \left\{ \frac{\delta f}{\delta x_1} \right\} + e_2 (1 - e_1) \right. \\ &\quad \left. \times \Pr \left\{ \frac{\delta f}{\delta x_2} \right\} + e_1 e_2 \Pr \left\{ \frac{\Delta f}{\Delta x_1 x_2} \right\} \right], \end{aligned} \quad (3)$$

where $\Pr\{\cdot\}$ = signal probability function and returns the probability of Boolean argument to be “1.”

Computation of $\Pr\{\cdot\}$ is briefly explained in [7]. For intermediate stages, output signal probability for each faulty gate is computed based on its input signal probabilities and reliability function. The process of computing output error probability and signal probability is continued until all gates are visited. After that, reliability for each individual primary output is determined. Based on the independency of the individual primary output signals [7], the desired circuit reliability is computed by multiplying its output reliabilities.

2.4. Probabilistic Transfer Matrix (PTM). Probabilistic transfer matrix (PTM) is a probabilistic modeling tool that performs instantaneous computation over all possible input combinations and calculates the exact probabilities of failures [8]. It is based on matrix representation where row indices represent output values and column indices represent input values. For an illustration purpose, PTM for a standard NOR gate is shown in (4) where P represents the probability for incorrect output value.

PTM representation can be extended to any logic gates. Other than a standard gate alone, PTM also looks into three different additional properties of a given circuit design such as independent wire, fan-out gates, and wire swaps. Independent wire has no error and is represented as 2 by 2 identity matrices, I . The output is the result of its input values with a probability of 1. A fan-out gate is denoted by F_n , where it feeds an input signal to its n outputs. Figure 1 shows the 2-output fan-out NAND gates which relates to the

FIGURE 1: 2-output fan-out NAND gates.

FIGURE 2: 2-wire swaps [11].

matrix representation shown in (5). Wire swaps are crossing wires. Figures 2 and 3 show 2-wire swaps and 3-wire swaps, respectively. In respective ways the wire swaps matrices are shown in (6) and (7).

For a given circuit design, PTM construction for its subcircuits and circuit is briefly explained in [8]:

$$\text{PTM} = \begin{bmatrix} P & 1-P & 1-P & 1-P \\ 1-P & P & P & P \end{bmatrix}, \quad (4)$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (5)$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (6)$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (7)$$

2.5. Bayesian Network (BN). BN is a graphical model whose basic elements are nodes and set of directed links between multiple nodes that creates a directed acyclic graph (DAG) [12]. The directed acyclic links are used to show joint probabilities between these nodes. Directed acyclic graph can be transformed into moral graph by adding undirected edges between the parents of a common child node and dropping the directions of the links. Moral graph can be transformed into triangulated graph if each of its cycle's length (i.e., greater than three) has a chord [13]. The triangulated graph thus formed enters into the stage junction tree graph. A directed acyclic graph is a junction tree graph, if each node that

FIGURE 3: 3-wire swaps [11].

FIGURE 4: Conceptual representation of the desired circuit.

belongs to two directed acyclic trees also belongs to every directed acyclic tree in the unique path between them [12, 13].

Each BN node (gate) has one conditional probability table, except their parent nodes. The node's conditional probability table is based on the truth table (gate type) of that node and probability of error, P . Each parent node has an earlier probability.

2.6. Error Probability Computation. Circuits can be represented by Bayesian Networks where inputs are the root nodes, outputs are the leaf nodes and internal signals are internal nodes [12, 13]. Reliability of the desired circuit is computed by assuming that the original circuit network is error-free and making a copy of it to represent an error prone network as shown in Figure 4. The shaded ones represent the error-free network whereby the nonshaded ones represent the error-prone network with gate probability of error, P . This conceptual representation of circuit can be represented by Bayesian network as shown in Figure 5. Both ideal and error free networks share the same inputs.

Error at the i th output can be represented mathematically in (8):

$$E_i = Y_i^e \otimes Y_i, \quad (8)$$

$$P(E_i = 1) = P(Y_i^e \otimes Y_i = 1),$$

where Y_i and Y_i^e are the outputs of the ideal and error prone circuits, respectively. The probability of the i th output can be calculated if both error free output Y_i and error prone Y_i^e are equal to logic 1. The XOR gate is used as a comparator to compare both the original and error prone networks which gives the correct reliability.

3. System Design

The automated tool is developed based on the generalization of four existing reliability models, namely, PGM, BDEC,

FIGURE 5: Bayesian network representation of the desired circuit.

FIGURE 6: C17.

PTM, and BN, but in this section, we use PTM based automated tool to compute PTM for subcircuits, circuit, and then reliability of C17 as shown in Figure 6. Input to the MATLAB-based tool is the Netlist of C17 in the form of its GPM, ACM, and GLM as shown in (9), (10), and (11), respectively. Traditionally, C17 is divided manually into several stages for PTM computation but with the developed automated tool, division of stages can be done automatically.

GPM signifies the types of logic gates in C17. The logic gates are represented by the first two alphabets. For example, NA represents NAND gate. OU represents C17 output signal. Integer at the back of the first two alphabets represents the number of logic gates/outputs of the same type in C17:

$$\begin{aligned} \text{GPM} \\ = [\text{NA1; NA2; NA3; NA4; NA5; NA6; OU1; OU2}] . \end{aligned} \quad (9)$$

ACM signifies the interconnection between the logic gates in C17. In the ACM matrix as shown in (10), a connection between logic gates is denoted by 1, and a no-connection is denoted by 0. From top to down, rows represent the input signals, types of logic gates, and output signals. From left to

right, columns represent the types of logic gates and output signals:

ACM

$$\begin{aligned} & \text{NA1 NA2 NA3 NA4 NA5 NA6 OU1 OU2} \\ & \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} \text{IN1} \\ \text{IN2} \\ \text{IN3} \\ \text{IN4} \\ \text{IN5} \\ \text{NA1} \\ \text{NA2} \\ \text{NA3} \\ \text{NA4} \\ \text{NA5} \\ \text{NA6} \\ \text{OU1} \\ \text{OU2} \end{matrix} . \end{aligned} \quad (10)$$

GLM signifies the layout matrix of logic gates (division of gates according to stages) in C17. In the GLM matrix shown in (11), the 1st integer 2 represents two gates in the first gates

FIGURE 7: How automated tool generates PTM and reliability of C17.

stage, the 2nd integer 2 represents two gates in the middle gates stage, and the 3rd integer 2 represents two gates in the last gates stage. The sum of the integers is the total number of logic gates in C17:

$$\text{GLM} = [2 \ 2 \ 2]. \quad (11)$$

A general graphical overview of how the developed automated tool computes PTM for subcircuits, circuit, and then reliability of C17 is shown in Figure 7. The tool first scans the Netlist of C17 to identify the inputs, the outputs, and the logic gates designation. In the next step, the tool divides up the circuit into different stages for PTM computation by implementing the front-tracking method from C17 input gates to its output gates. For C17, a total of six stages are identified by the tool before it starts to create PTMs.

For Stage 1, Stage 3, and Stage 5 in Figure 6, the tool creates input-output mapping matrix (IOM). It describes the relationship between the input signals and the output signals in that stage. A stage with c output signals results in an IOM of dimensions $c \times 2$ where the 1st column represents the input signals and the 2nd column represents the output signals in that stage.

For C17 (Stage 1), input signals (1st column) and output signals (2nd column) form the IOM as shown in (12). The n counts of the same input signals signify the n fanouts from the input signals whereby the space between the same input signals signifies the crossover of wire(s).

For Stage 2, Stage 4, and Stage 6, the tool uses MATLAB commands for calculating Kronecker products. If A is a $c \times d$ matrix and B is a $e \times f$ matrix, then the Kronecker product of A and B is a matrix of dimensions $ce \times df$.

4. Simulation Results and Discussions

Stage 1. This wires-only stage has a set of fanouts that increases the number of signals from 5 to 6. Its IOM matrix is shown in

(12). The IOM matrix is used to create a PTM matrix for Stage 1 that has $2^6 = 64$ input rows and $2^5 = 32$ output columns.

Stage 2. This wires-and-gates stage has two 2-input NAND gates and 2 single wires. In this stage, PTM is generated by the Kronecker product between the PTMs of the NAND gates and the wires. PTMs corresponding to the NAND gates and the wires are 4 by 2 and 2 by 2 matrices, respectively. The PTM matrix for Stage 2 has a dimension of 64×16 for gate error probability, $P = 0$. The PTM is not shown in this paper as it takes so much of space.

Stage 3. This wires-only stage has a fanout that increases the number of signals from 4 to 5. The corresponding IOM matrix is shown in (13). The IOM matrix is used to create a PTM matrix for Stage 3 that has $2^5 = 32$ input rows and $2^4 = 16$ output columns.

Stage 4. This wire-and-gates stage has two 2-input NAND gates and 1 single wire. In this stage, PTM is generated by the Kronecker product between PTMs of the NAND gates and the wire. The PTM matrix for Stage 4 has a dimension of 32×8 for gate error probability, $P = 0$.

Stage 5. This wires-only stage has a fanout that increases the number of signals from 3 to 4. The corresponding IOM matrix is shown in (14). The IOM matrix is used to create a PTM matrix for Stage 5 that has $2^4 = 16$ input rows and $2^3 = 8$ output columns.

Stage 6. This stage has two 2-input NAND gates. In this stage, PTM is generated by the Kronecker product between PTMs of the NAND gates. The PTM matrix for Stage 6 has a dimension of 16×4 for gate error probability, $P = 0$,

$$\text{IOM} = \begin{bmatrix} 1 & 1 \\ 2 & 3 \\ 3 & 2 \\ 3 & 4 \\ 4 & 5 \\ 5 & 6 \end{bmatrix}, \quad (12)$$

FIGURE 8: Series of NAND gates for computing its exact error threshold.

$$\text{IOM} = \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ 3 & 3 \\ 3 & 4 \\ 4 & 5 \end{bmatrix}, \quad (13)$$

$$\text{IOM} = \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ 2 & 3 \\ 3 & 4 \end{bmatrix}. \quad (14)$$

After computing PTMs for individual stages, the programmed tool generates PTM for C17 by multiplying PTMs of all individual stages for $P = 0$ as shown in (15).

Now, to compute reliability of C17 for $P = 0.05$, user has to run the tool twice. In the first run, $P = 0$, and in the second run, the desired value of $P = 0.05$ is used. The two runs produce two different PTMs. From the first PTM as shown in (15), the cells containing 1's are identified, followed by the summation, R of the corresponding cells in the second PTM as shown in (16).

Finally, reliability measure for C17 is computed as follow: $R/n = 25.0852/32 = 0.7839 = 78.39\%$, where n is number of rows in the PTM of C17.

To confirm the correctness of the automated tool based on PGM, BDEC, PTM, and BN techniques, reliability for a set of standard benchmark test circuits has been generated and verified that the automatically generated reliabilities matched the manually generated ones. Table 2 shows reliability measures, time execution, and storage complexity for standard benchmark test circuits with gate error probability, $P = 0.05$. The benchmark test circuits that have been tested with the developed tool apart from C17 are

- (i) full adder with 5 gates,
- (ii) NAND-based full adder with 9 gates,
- (iii) majority gates-based full adder with 28 gates,
- (iv) 2 to 4 decoder with 5 gates.

Consider the following:

$$\text{PTM} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (15)$$

5. Exact Error Threshold for Individual Faulty Gates by PTM

Using built-in feature of PTM, it can be hypothesized that the reliability of the desired circuit does not depend only on its faulty gates but also on the exact error threshold of these faulty gates above which no reliable computation is

possible. For this purpose, the automated tool is employed again to compute the exact error thresholds for individual faulty gates. In this approach, Von Neumann model has been considered since the faults result in the worst scenario for circuit's reliability [4–6]. According to the model, the worst case reliability is achieved when two input signals of any gate are equally probable to be “1.”

To determine exact error threshold for NAND gate, the automated reliability tool is employed on a series of NAND gates as shown in Figure 8. For a fixed gate error probability, $0 < P \leq 0.5$ and initial value of input signal being “1,” $0 \leq X1 \leq 1$, probability of output signal being “1” is computed for each stage until to a sufficiently large number of stages as shown in Figure 8. This process of iteration is continued until the probability of output signal being “1” converged to some fixed attractors. The plotting of those fixed attractors against each value of P is shown in Figure 9.

By applying the same procedure on a series of other gates, one can obtain similar kind of graph plots as shown in Figures 10, 11, and 12 for NOR, XOR, and NOT gates, respectively,

$$PTM = \begin{bmatrix} 0.7782 & 0.0841 & 0.0841 & 0.0537 \\ 0.1166 & 0.7456 & 0.0146 & 0.1231 \\ 0.7782 & 0.0841 & 0.0841 & 0.0537 \\ 0.1166 & 0.7456 & 0.0146 & 0.1231 \\ 0.7782 & 0.0841 & 0.0841 & 0.0537 \\ 0.1166 & 0.7456 & 0.0146 & 0.1231 \\ 0.0818 & 0.0109 & 0.7804 & 0.1268 \\ 0.0124 & 0.0804 & 0.1189 & 0.7884 \\ 0.0801 & 0.0512 & 0.0512 & 0.8176 \\ 0.0452 & 0.0860 & 0.0475 & 0.8213 \\ 0.0801 & 0.0512 & 0.0512 & 0.8176 \\ 0.0452 & 0.0860 & 0.0475 & 0.8213 \\ 0.0801 & 0.0512 & 0.0512 & 0.8176 \\ 0.0452 & 0.0860 & 0.0475 & 0.8213 \\ 0.0104 & 0.0438 & 0.1208 & 0.8249 \\ 0.0068 & 0.0475 & 0.0860 & 0.8597 \\ 0.7782 & 0.0841 & 0.0841 & 0.0537 \\ 0.1166 & 0.7456 & 0.0146 & 0.1231 \\ 0.7782 & 0.0841 & 0.0841 & 0.0537 \\ 0.1166 & 0.7456 & 0.0146 & 0.1231 \\ 0.7782 & 0.0841 & 0.0841 & 0.0537 \\ 0.7434 & 0.1189 & 0.0804 & 0.0574 \\ 0.0818 & 0.0109 & 0.7804 & 0.1268 \\ 0.0781 & 0.0146 & 0.7456 & 0.1616 \\ 0.0801 & 0.0512 & 0.0512 & 0.8176 \\ 0.0452 & 0.0860 & 0.0475 & 0.8213 \\ 0.0801 & 0.0512 & 0.0512 & 0.8176 \\ 0.0452 & 0.0860 & 0.0475 & 0.8213 \\ 0.7415 & 0.0823 & 0.0823 & 0.0939 \\ 0.7396 & 0.0842 & 0.0821 & 0.0941 \\ 0.0780 & 0.0127 & 0.7457 & 0.1635 \\ 0.0779 & 0.0129 & 0.7439 & 0.1654 \end{bmatrix} \cdot \quad (16)$$

Based on Figures 9 and 10, both NAND and NOR faulty gates share the same error threshold which is <0.1 . This phenomenon happened because the NOR gate is a duality of the NAND gate, so their stationary behaviors, characterized by Figures 9 and 10, appear to be complementary and they

FIGURE 9: Graph plot for NAND gate.

FIGURE 10: Graph plot for NOR gate.

share the same error bound. For XOR gate in Figure 11, its maximum error threshold is 0, or in other words a maximum error threshold never occurs for this gate. This denotes that if a circuit system is only made of XOR gates, its output always becomes irrelevant to its inputs. For NOR gate in Figure 12, its maximum error threshold is $<3 \times 10^{-3}$.

6. Conclusion

Reliability analysis is becoming very time consuming and troublesome as the computational complexity continues to increase exponentially with circuit size. In this paper, firstly, we discussed the development of an automated reliability evaluation tool that has the ability to speed up the reliability evaluation process for nanotechnology-based circuit systems. It is developed based on the generalization of PGM, BDEC, PTM, and BN. Input to the developed MATLAB-based tool is the desired circuit Netlist in the form of its GPM, ACM,

TABLE 2: Reliability measures, time execution, and storage complexity for standard benchmark test circuits with $P = 0.05$.

Test circuit	Number of faulty gates	Model	Reliability	Time (s)	Storage (bytes)
Full adder	5	PGM	0.7879	4.46	1962
		BDEC	0.7875	0.61	802
		PTM	0.7997	6.56	173872
		BN	0.8869	3.33	108654
2-4 decoder	6	PGM	0.7397	1.55	1836
		BDEC	0.7405	0.69	1260
		PTM	0.7566	41.38	1201826
		BN	0.9166	6.18	131642
CI7	6	PGM	0.7621	9.10	7088
		BDEC	0.7634	0.55	1096
		PTM	0.7839	6.27	142102
		BN	0.8816	3.13	139456
NAND based full adder	12	PGM	0.5933	4.16	3348
		BDEC	0.6326	0.88	2188
		PTM	0.6605	11.33	327268
		BN	0.7809	4.13	227444
Majority gates based full adder	28	PGM	0.5767	8.27	9460
		BDEC	0.6274	1.94	8300
		PTM	Storage complexity increases result in “busy” mode		
		BN			
			0.7560	7.55	594548

FIGURE 11: Graph plot for XOR gate.

FIGURE 12: Graph plot for NOT gate.

and GLM. Secondly, we carried out a comparative study involving reliability measures, time execution, and storage complexity on a set of standard benchmark test circuits using the developed automated tool. Finally, the paper is extended to determine the exact error threshold for individual faulty gates above which no reliable computation is possible. Apart from faulty gates, error threshold in those faulty gates is significantly important subject to be considered in designing nano-based circuits. Future work will look into application of PTM-based automated tool for larger circuits with ≥ 28 faulty gates.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The author is grateful to Universiti Teknologi PETRONAS for its financial support.

References

- [1] D. T. Franco, M. C. Vasconcelos, L. Naviner, and J.-F. Naviner, "Signal probability for reliability evaluation of logic circuits," *Microelectronics Reliability*, vol. 48, no. 8-9, pp. 1586–1591, 2008.
- [2] R. I. Bahar, J. Mundy, and J. Chen, "A probabilistic-based design methodology for nanoscale computation," in *Proceedings of the International Conference on Computer Aided Design (ICCAD '03)*, pp. 480–486, November 2003.
- [3] J. Han and P. Jonker, "A defect- and fault-tolerant architecture for nanocomputers," *Nanotechnology*, vol. 14, no. 2, pp. 224–230, 2003.
- [4] J. Han, E. Taylor, J. Gao, and J. Fortes, "Faults, error bounds and reliability of nanoelectronic circuits," in *IEEE 16th International Conference on Application-Specific Systems, Architectures, and Processors (ASAP '05)*, pp. 247–253, July 2005.
- [5] J. B. Gao, Y. Qi, and J. A. B. Fortes, "Bifurcations and fundamental error bounds for fault-tolerant computations," *IEEE Transactions on Nanotechnology*, vol. 4, no. 4, pp. 395–402, 2005.
- [6] J. Han, E. Taylor, J. Gao, and J. Fortes, "Reliability modeling of nanoelectronic circuits," in *Proceedings of the 5th IEEE Conference on Nanotechnology*, pp. 269–272, July 2005.
- [7] N. Mohyuddin, E. Pakbaznia, and M. Pedram, "Probabilistic error propagation in logic circuits using the boolean difference calculus," in *Proceedings of the 26th IEEE International Conference on Computer Design*, pp. 7–13, October 2008.
- [8] S. Krishnaswamy, G. F. Viamontes, I. L. Markov, and J. P. Hayes, "Accurate reliability evaluation and enhancement via probabilistic transfer matrices," in *Proceedings of the Design, Automation and Test in Europe (DATE '05)*, vol. 1, pp. 282–287, March 2005.
- [9] K. Patel, I. Markov, and J. Hayes, "Evaluating circuit reliability under probabilistic gate-level fault models," in *Proceedings of the International Workshop on Logic Synthesis*, pp. 59–64.
- [10] W. Ibrahim, V. Beiu, and M. H. Sulieman, "On the reliability of majority gates full adders," *IEEE Transactions on Nanotechnology*, vol. 7, no. 1, pp. 56–67, 2008.
- [11] A. Beg and W. Ibrahim, "On teaching circuit reliability," in *Proceedings of the 38th Annual Frontiers in Education Conference*, Saratoga Springs, NY, USA, October 2008.
- [12] T. Rejimon and S. Bhanja, "Scalable probabilistic computing models using Bayesian networks," in *Proceedings of the 48th Midwest Symposium on Circuits and Systems*, pp. 712–715, Covington, Ky, USA, August 2005.
- [13] F. Taroni and C. Aitken, *Bayesian Networks and Probabilistic Inference in Forensic Science*, Statistics in Practice, 2006.