Microelectronics Reliability 55 (2015) 1285–1289

Contents lists available at ScienceDirect

Microelectronics Reliability

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

A novel analytical method for defect tolerance assessment

M. Slimani *, A. Ben Dhia, L. Naviner

Institut Télécom/Télécom ParisTech, CNRS-LTCI UMR 5141 Paris, France

ARTICLE INFO

Article history:
Received 21 May 2015
Accepted 12 June 2015
Available online 4 July 2015

Keywords:
Fault tolerance analysis
Analytical methods
Error propagation
Analog fault simulation

ABSTRACT

Due to technology downscaling, defect tolerance analysis has become a major concern in the design of digital circuits. In this paper, we present a novel analytical method that calculates the defect tolerance of logic circuits using probabilistic defect propagation. The proposed method is explained in case of single defect model, but can be easily adapted to handle multiple fault scenarios. The approach manages signal dependencies due to reconvergent fanouts, providing accurate results and performing simple operations.

1. Introduction

As technology scales down to the nanometer era, reliability of integrated circuits is rapidly becoming a major concern in the design of electronic circuits [1,2]. Evaluating the impact of possible faults on the circuit functionality at an early stage of the design flow is highly important to make judicious choices in the design hardening before the fabrication process. Hence, many approaches have been proposed for reliability analysis [3,5–7]. They can be classified into two main categories: simulation based-methods and analytical methods.

Analytical fault tolerance analysis approaches suffer from either accuracy or scalability problems. The Probabilistic Transfer Matrix (PTM) approach, introduced in [4], is among the most known analytical methods. Despite its accuracy, its complexity grows exponentially with the number of inputs and outputs, leading to an intractable computation time and a need for a big storage space in memory, even for medium-sized circuits [8]. The Signal Probability Reliability (SPR) method, introduced in [9] outperforms the PTM approach thanks to its linear complexity with the number of logic gates in the circuit. Nonetheless, the more reconvergent fanouts there are in the circuit, the more SPR loses accuracy in computing the circuit reliability. The SPR-Multi Pass (SPR-MP) method has been then proposed to enhance the original SPR algorithm by tackling the problem of reconvergent fanouts [10]. Indeed, SPR-MP takes into account the correlation of signals by performing the analysis in multiple passes. A single state of the fanout signal is considered in each pass. Thus, there are 4 possible passes for each fanout, corresponding to four partial reliability values that should be added in the end. Like the PTM approach, SPR-MP is an accurate algorithm. However, its complexity grows exponentially with the number of reconvergent fanouts in the circuit.

In this paper, we propose an analytical approach that provides accurate results while performing simple operations. In Section 2, we introduce the basic concept of the method to compute the circuit failure rate from those of gates. Then, more complicated scenarios dealing with fanouts and reconvergent fanouts are considered. Besides, the method uses realistic gates failure rates extracted through transistor-level fault simulation. So, in Section 3, transistor defect tolerance analysis is presented. Simulation results are discussed in Section 4. Finally, concluding remarks are drawn in Section 5.

2. Proposed error propagation method

Let G = $(g_1, g_2, \ldots\ldots, g_n)$ be an implementation of a given combinational circuit synthesized in a standard cell library. Let $FR(g_i)$ be the failure rate of the circuit when the defect takes place at the gate g_i. As we deal with a single defect model, the global failure rate of the circuit is the average failure rate expressed in (1) where n is the number of the gates constituting the circuit.

$$F = \frac{1}{n}\sum_{i=1}^{n} FR(g_i) \tag{1}$$

Here, $FR(g_i)$ corresponds to the probability that at least one of the circuit outputs is incorrect due to a transistor-level defect affecting the gate. This means, first that the defect affected the output of the gate itself and second that this fault reached the output of the circuit without being logically masked. These two phases are totally independent and could be classified into transistor and gate-level propagations.

[*] Corresponding author.
E-mail address: mariem.slimani@telecom-paristech.fr (M. Slimani).

http://dx.doi.org/10.1016/j.microrel.2015.06.059

Therefore, the failure rate $FR(g_i)$ can certainly be written as the product of the failure rates due to transistor and gate level propagations.

Transistor-level propagation results in altering the gate output because of a defect occurring within the gate. Let us define FR^i the probability that the output of the gate g_i fails due to an inner defect. Two cases are to be considered: either the fault-free output value is a logic '0' and the obtained output is at logic '1' or the correct output value is '1' and the output is inverted. We define $(FR^{01}, FR^{10})^i$ the couple of probabilities that the output of the gate g_i is incorrect associated to the couple of inversion errors ('0' → '1', '1' → '0'). We will describe in Section 3 how these parameters can be extracted using a transistor-level fault analysis. We characterize each library cell with the defect vector DF depicted in (2).

$$DF = \begin{bmatrix} FR^{01} & FR^{10} \end{bmatrix} \tag{2}$$

At gate level, the defect is modeled by the stuck-at fault model, where a net is stuck at the logical value '0' or '1'. The probabilities that the stuck-at error affects the output of the next gate depend on the logic function of that following gate and its inputs probabilities. For instance, consider the NAND gate shown in Fig. 1. The stuck-at-1 error at the input A results in a '1' → '0' inversion at the output S when the input B is at logic value '1'. Similarly, the output S is incorrect with a '0' → '1' inversion when a stuck-at-0 error affects the input A and B is at logic value '1'.

Let us define $(FR^{01}_{st0}, FR^{10}_{st0}, FR^{01}_{st1}, FR^{10}_{st1})$ the probabilities that the output of the gate is incorrect and the error is either '0' → '1' or '1' → '0' inversion due to a stuck-at-0 or to a stuck-at-1 fault at one of its inputs. We characterize each logic function with the Failure Matrix (FM) shown in (3).

$$FM = \begin{bmatrix} FR^{01}_{st1} & FR^{10}_{st1} \\ FR^{01}_{st0} & FR^{10}_{st0} \end{bmatrix} \tag{3}$$

The FM can be easily extracted from the truth table of the logic function. For the NAND gate, where we consider the fault on the input A, we deduce that $FR^{01}_{st0} = FR^{10}_{st1} = p(B=1)$. FR^{10}_{st0} and FR^{01}_{st1} are both equal to '0' as the output of the NAND gate can never make a '1' → '0' or '0' → '1' inversion when the error is a stuck-at-0 or stuck-at-1, respectively.

Let us take another example. By looking at the truth table of the XOR gate, we can observe that when a stuck-at-1 error affects one of its inputs, the output can fail with a '0' → '1' inversion when the correct input, suppose B, is at logic value '0' and with '1' → '0' inversion when the correct input is at logic value '1'. Likewise, when the error is a stuck-at-0, the output can fail with a '1' → '0' or '0' → '1' inversion when the correct input is at logic value '0' or '1', respectively. Hence, the FM of the XOR gate can be written as in (4).

$$FM_{XOR} = \begin{bmatrix} p(B=0) & p(B=1) \\ p(B=1) & p(B=0) \end{bmatrix} \tag{4}$$

Actually, the FM carries the information whether the defect at one of the input of the gate manages to propagate to its output. Hence, by multiplying the FMs of gates in series, we can see whether the defect at the input of the first gate reaches the output of the last one.

Taking into account the failure rates of the defective gate $(FR^{01}, FR^{10})^i$ and the FMs of gates belonging to different paths relating the defective gate to the outputs of the circuit, we could get the failure rate of the circuit when the defect takes place at the gate g_i. The processing performed depends on the nature of these paths. In the next paragraphs, different examples will be used to explain how to compute the failure rate in different scenarios with increased order of complexity. Note that the obtained result is always the same as the one given by the exact method SPR-MP.

2.1. Example 1: simple path with gates in series

Let us consider the example depicted in Fig. 2.

Assume that the output S_1 is faulty due to a defect that takes place in g_1. The parameters of the XOR DF vector (FR^{01} and FR^{10}) corresponding to g_1 have been extracted from analog fault simulation. The defect at S_1 can alter the output S_5 if it passes through gates (g_3 and g_5) without being logically masked. Multiplying the FMs of the gates g_3 and g_5, we get a global FM indicating the probabilities that the defect reaches S_5. We would express the failure rate of the gate g_1 as in (5), where DF_{g_1} is the DF of the XOR gate, FM_{g_3} is the FM of the XOR gate while $B = S_2$ and FM_{g_5} is the FM of the NAND gate with $B = S_4$.

$$FR(g_1) = \sum \left[DF_{g_1} \times FM_{g_3} \times FM_{g_5} \right] \tag{5}$$

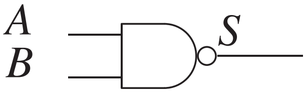
2.2. Example 2: path with one fanout

Fig. 3(a) shows an example with one fanout. The error occurring at the node F_1, due to a physical defect at the gate g_1, can alter the output S_1 by propagating through the path $P1 = F_1 g_2 g_3 S_1$. It can also induce an error in S_2 if it is not masked by any gates in path $P_2 = F_1 g_4 g_5 S_2$. We would express the failure rate $FR(g_1)$ as in (6), where FR_{P_1} and FR_{P_2} are the failure rates of the circuit due to the propagation of the defect through P_1 and P_2, respectively. $FR_{P_1 \cap P_2}$ is the joint failure rate corresponding to the case where the defect reaches both outputs S_1 and S_2.

$$FR(g_1) = FR_{P_1} + FR_{P_2} - FR_{P_1 \cap P_2} \tag{6}$$

FR_{P_1} and FR_{P_2} are calculated following Example 1. To calculate the joint failure rate, we multiply the FMs of gates in each path from the fanout node to the outputs. For instance, let $FM_{P_1} = FM_{g_2} \times FM_{g_3}$ and $FM_{P_2} = FM_{g_4} \times FM_{g_5}$ be these products. Obviously, like any other FM, the sum of terms in the first row and the sum of those in the second row are the failure rates due to the stuck-at-1 and stuck-at-0 fault, respectively. Tensoring product the sum of terms in the first rows of FM_{P_1} by that of FM_{P_2}, we get the joint failure rate due to stuck-at-1 error (FR_1). In the same way, we tensor product the sum of terms in the second rows of the same matrices to get the joint failure rate due to stuck-at-0 error (FR_0). Finally, $FR_{P_1 \cap P_2}$ can be expressed as:

$$FR_{P_1 \cap P_2} = DF_{g_1} \begin{bmatrix} FR_1 \\ FR_0 \end{bmatrix}. \tag{7}$$



A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

Fig. 1. Example of fault propagation.

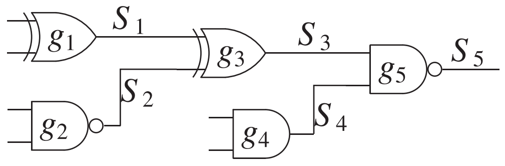


Fig. 2. Fault propagation through gates in series.