Modeling the Reliability of Digital Operators

Hicham Ezzat

Université Pierre et Marie Curie

Mastère de Sciences et Technologies

Mention Informatique - Spécialité ACSI
2008 – 2009

Spécialité : Architecture et Conception
des Systèmes Intégrés

MODÉLISATION DE LA FIABILITÉ D'OPÉRATEURS NUMÉRIQUES

RAPPORT DE SOUTENANCE

Avril 2009

PRÉSENTÉ PAR

HICHAM EZZAT

ENCADRANTE

LÍRIDA NAVINER

Etablissement d'Accueil : Télécom ParisTech
Département COMELEC
Equipe SEN (Système Electronique Numérique)

Contents

List of Figures

Chapter 1

Présentation du stage de recherche

1.1 L'Institution d'acceuil

1.1.1 Organisation générale

TELECOM ParisTech appartient à l'Institut TELECOM mais est aussi membre du pôle ParisTech. L'Institut TELECOM, regroupe 6 grandes écoles dont les disciplines d'enseignement et de recherche sont principalement axées sur les sciences et technologies de l'information. Le pôle ParisTech, lui, en regroupe maintenant onze, mais ses domaines sont plus vastes et couvrent pratiquement toutes les sciences de l'ingénieur. Le groupe ParisTech est lui même membre du réseau IDEA League qui rassemble cinq différents pôles de recherche européens. Dans des approches interdisciplinaires plus applicatives, au sein de projets de recherche collaboratifs et dans la réalisation de contrats industriels, la Recherche de TELECOM ParisTech ambitionne de relever les défis de la société de l'information émergente.

Partenaire de taille de l'université Pierre et Marie Curie, entre autre au sein de l'École Doctorale d'Informatique, Télécommunications et Électronique (EDITE) de Paris, TELECOM ParisTech regroupe environ 400 chercheurs de plusieurs catégories (doctorants, enseignants, etc.) Sa formation d'ingénieur peut être complétée par un master orienté recherche et un diplôme de doctorat ou par un master spécialisé. Il est aussi possible d'y suivre des enseignements dans le cadre de formation continue. Les différents axes de recherche à TELECOM ParisTech se répartissent sur les départements suivants:

- Informatique & Réseaux (INFRES)
- Traitement du Signal et des Images (TSI)
- Économie, Gestion, Sciences Humaines & Sociales (EGSH)
- Communications & Électronique (COMELEC)

1. **Informatique & Réseaux (INFRES) :**

   La fertilisation mutuelle de l'informatique et des réseaux doit prendre en compte la dimension multiservice des systèmes pour améliorer la performance des communications de plus en plus diversifiées. La recherche du département INFRES répond à ces nouveaux défis : urbanisation numérique pour réseaux hétérogènes, modélisation et ingénierie de la connaissance, nouveaux paradigmes pour l'informatique et les réseaux du futur.

2. **Traitement du Signal et des Images (TSI) :**

   Ce département se concentre sur l'étude de l'image sous toutes ses formes : numérique, optique... pour des applications variées : médicale, satellitaire, artistique... Mais dans ses principaux thèmes de recherche se trouvent aussi l'étude de la parole, du son ainsi que les problèmes de codage et de transmission de ces différentes informations.

3. **Économie, Gestion, Sciences Humaines & Sociales (EGSH) :**

   Les études et les recherches portent sur les Technologies d'Information et de Communication (TIC) selon diverses problématiques sociologiques, linguistiques, ergonomiques, cognitives, économiques, de gestion... Il s'agit principalement d'analyser les rapports qu'entretiennent les TIC.

4. **Communications & Électronique (COMELEC) :**

   Au conluent des activités "composants", "communications", et "services", ce département regroupe l'ensemble des vecteurs technologiques de l'informatique multimédia. Il assure l'interface entre les derniers concepts algorithmiques et les technologies de pointe de l'optique et de l'électronique pour réaliser le transport et le traitement de l'information à l'aide des techniques les plus en amonts. C'est au sein de ce département COMELEC que se réalise le stage.

1.1.2 Le Laboratoire d'acceuil

L'association entre communications et électronique facilite la définition des actions de recherche, déterminantes pour la contribution au programme doctoral de l'École et pour l'appartenance de l'ensemble du département à l'Unité de Recherche associée au CNRS (UMR 5141 LTCI). COMELEC regroupe près de 45 personnels permanents, essentiellement chercheurs ou enseignants-chercheurs, et quelques administratifs et techniciens. Il accueille en permanence près de 80 doctorants et ouvre ses laboratoires à environ 30 stagiaires chaque année.

Le département COMELEC est lui-même divisé en huit groupes de recherche :

- Systèmes et Électroniques Numériques (SEN)
- Systèmes Intégrés Analogiques et Mixtes (SIAM)
- Communications Numériques (COMNUM)
- Optoélectronique et Communications Optiques (GTO)
- Radiofréquences et Micro-ondes (RFM)
- Exploration des d'Architectures de Systèmes Intégrés (labSoc)
- SMART qui réunit les administratifs et techniciens du département (SMART)

La recherche de l'équipe SEN, elle, gravite autour des architectures des systèmes embarqués communicants sécurisés. Elle est orientée par deux thèmes principaux : d'une part , les architectures électroniques des systèmes embarqués en vue de leur intégration efficace sur puce ou sur carte ; d'autre part la sécurisation des circuits contre les attaques par canaux cachés (attaques en analyse de l'activité du circuit ou en injection de faute).

1.2 Le thème du stage

L'évolution régulière des finesses de gravure en microéléctronique (loi de Moore) joue un role fondamental dans le dévelopement économiques depuis une cinquantaine d'années. Cependant, les technologies décananométriques conduisent désormais à des rendements nettement inférieurs à ceux obtenus avec les générations précédentes du fait de la proximité des limites physiques. Cette baisse de rendement et une variabilité paramétrique fortement accrue entrainent une moindre fiabilité des systèmes concus pourtant fondamentale pour de nombreuses applications (médical, transports, spatial, etc...). De tels inconvénients seront également présents dans de futures nantechnologies alternatives ou complémentaires. Ce stage porte sur la modélisation de la fiabilité d'opérateurs numériques.

Chapter 2

Introduction of the Topic

2.1 Study Context

Integrated circuits have known a constant evolution in the last decades, with increases in density and speed that follow the rates predicted in Moore's law [1]. Figure 2.1 shows the evolution of density in ICs allowed by technology scaling. This increase in circuits density is referred as an improvement in the number of transistors in an IC.

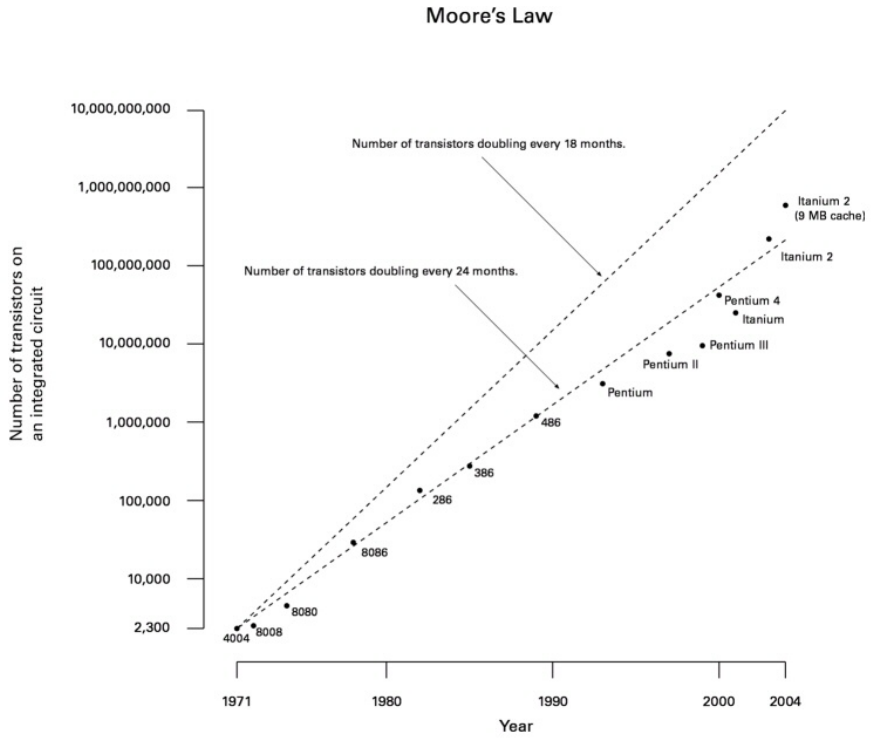

Figure 2.1: Moore's Law [https://www.vpac.org/node/242].

Several measures of digital technology (size, cost, density and speed of compo-

nents) are also improving at exponential rates related to this law. The most popular formulation of Moore is of the doubling of the number of transistors on integrated circuits every two years. At the end of the 1970s, Moore's law became known as the limit for the number of transistors on the most complex chips. Recent trends show that this rate has been maintained into the year 2007.

The basic principle behind this evolution is the reduction, or scaling, in the dimensions of the integrated structures. The willpower and possibility to create very small devices enables them to work faster, makes them cheaper to manufacture and allows improvements in processing power. In fact, the progress of the CMOS technology is an important driver for telecommunications evolution, with the continuous integration of complex functions needed by demanding applications. As integrated circuits evolve, they approach some limits that make further improvements more difficult and even unpredictable. With reduced voltages and increased speed and density, the reliability of deep-submicron circuits is a new concern for designers, since noise immunity is reduced and thermal noise effects show-up.

The main problematic comes from the fact that in new Nanotechnology, CMOS devices will have a higher probability of being defective than current ones due to the variability and physical rules of their fabrication processes. As a result, in circuits fabricated using these devices, reliability will be a key design concern.

The reliability of integrated circuits has thus become a key consideration when dealing with nanoscale designs, as a consequence of many factors associated with technology scaling, like manufacturing precision limitations, devices parametric variations, supply voltage reduction, higher operation frequency and power dissipation concerns as shown in [2]. These problems are a serious menace to the continuous evolution observed in the development of the integrated circuits industry. There exist many techniques to improve or to counteract the reduction of reliability in integrated circuits but, generally, these techniques reduce the gains achieved with scaling and there will be a point where scaling will be meaningless. The problem in determining this point is the complexity of reliability evaluation, that leads to the use of probabilistic and stochastic methods. The reliability of a circuit is dependent on too many variables and the growing complexity of the circuits themselves does not make the task easier.

As revealed by Way Kuo in his article in [3], to Build for the Future, we must achieve major advances related to reliability in addition to exploring and discovering interdisciplinary connections in important cutting-edge research areas. The technologies for today's design and manufacturing have for some time been steadily moving from the realm of the micro-scale to the nano-scale, but advancements in reliability have not kept up with the pace!

This work is part of the research project called NanoElec, which deals with the study of new electronic issues. One objective of this project is the development of

methodologies to design reliable processors to face the declination of new (nano) technology circuits' reliability. We present the state of the art in proposed solutions and alternatives that can improve the chances of a large utilization of these nanotechnologies.

The integration of this reliability analysis in the design-flow of any logic circuit is certainly an important step in the direction of a reliability-aware design process.

The main objectives of the work are

- Contribute to the study of compromises strength-speed-cost of self-testable operators already developed by NanoElec team in the Télécom ParisTech laboratory.
- Implement a tool for computing the reliability of digital circuits that could be integrated to the design flow of any logic circuit.

2.2 Definition and Problem Analysis

2.2.1 Reliability Analysis Definition

The interest of the work is targeted to the useful life period of the bathtub curve of Figure 2.2, where fault occurrence is highly related to random nature sources. This eliminates the reliability analysis concerning infant mortality, that is related to manufacturing issues, and reliability analysis concerning wearout mechanisms like electromigration, hot carriers, time-dependent dielectric breakdown, among others.

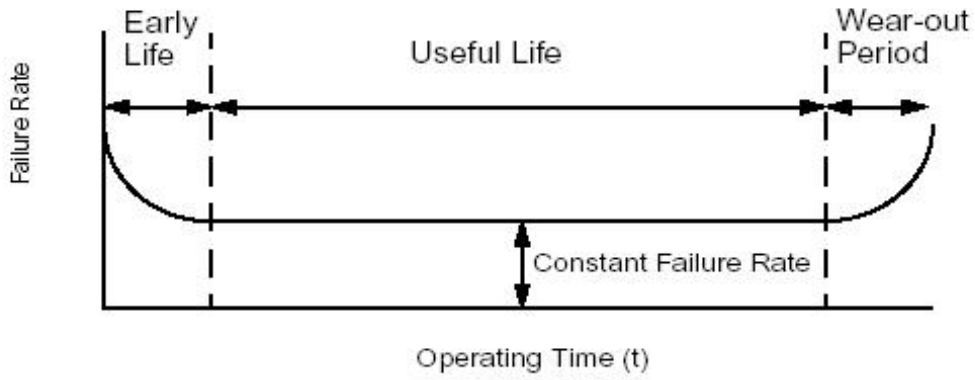

Figure 2.2: The bathtub curve

The following subsection presents chronologically a revue on several works concerning reliability analysis of logic circuits, presenting their objectives, advantages and disadvantages and allowing a characterization of the proposed solutions and a definition of the focus of the current work.

2.2.2 State of the Art on Reliability Analysis

1. **Probabilistic Transfer Matrix (PTM)**

   The works of Krishnaswamy [4] and Patel [5] present a classical straightforward methodology to compute the reliability of logic circuits. The reliability analysis is based on what is called a Probabilistic Transfer Matrix (PTM) specified for each gate of the logic circuit, which models the signal probability at the output according to the probability of inputs and the gate reliability value. A fault-free gate, i.e. a gate with a reliability equal 100%, is then represented by a PTM matrix with a gate reliability value 100%, wich is called an Ideal Transfer Matrix (ITM).

   Figure 8.1 shows an example of the available ITM and PTM for an NAND2 logic gate, where q is the NAND2 logic gate reliability value and $1 - q$ is the NAND2 logic gate probability of error.

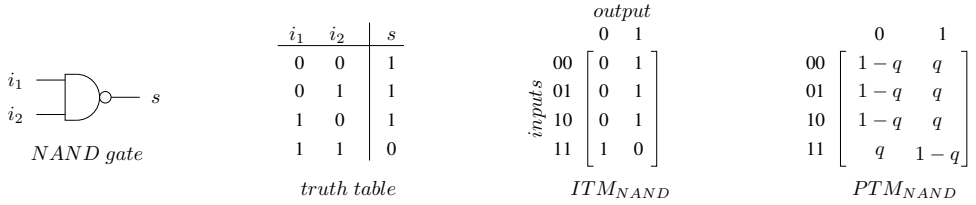

Figure 2.3: PTM and ITM representation of an NAND2 logic gate

The reliability of the whole circuit is simply determined by the circuit PTM and the circuit ITM. The kronecker product of the PTMs of the elements of a level represent the PTM of the corresponding level, while the product of all levels PTMs represent the total circuit PTM. Figure 8.2 shows an example of PTM computation for the circuit C17 from ISCAS'85 Benchmark. Having the matrix of the circuit PTM, the circuit reliability is simply obtained by adding the probabilities of the circuit PTM corresponding to a logic "1" in the circuit ITM.

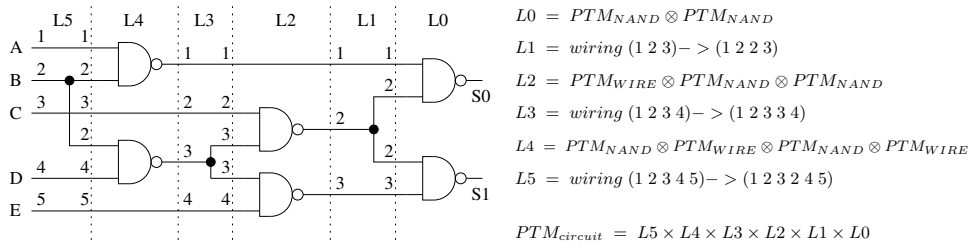

Figure 2.4: PTM of the C17 from ISCAS'85 Benchmark

The main advantages of the PTM model are its straightforward implementation. The main disadvantage is the scalability problem since the PTM of a circuit is a matrix representation of input and output pattern possibilities, so the size of

this matrix increases exponentially with the number of inputs and outputs, what leads to intractable computing times and memory storage needs for practical circuits. This limits a lot the processing and demands a huge storage (memory) capacity, being useful only for small circuits. This PTM reliability analysis method is thus difficult to apply in a Nanotechnology context where nanoscale circuits have millions to billions of signals. Despite the scalability problems of this PTM model, evaluation tools have been implementated in order to exploit the characteristics of the method and verify its practical limits.

2. **Probabilistic Gate Model (PGM)**

   Han et al. [6] presented the Probabilistic Gate Model (PGM) of individual gates to iteratively build the reliability of its output. The PGM-based methodology needs the formulation of a PGM for each logic gate type. For instance, the PGM for the NAND2 gate of Figure 8.1 with a failure probability of ϵ is as in the equation in 2.1

$$p(out) = \begin{bmatrix} 1 - p(A_0)p(B_0) & p(A_0)p(B_0) \end{bmatrix} . \begin{bmatrix} 1-\epsilon \\ \epsilon \end{bmatrix} \tag{2.1}$$

   where $p(A_0)$ and $p(B_0)$ are the probabilities of inputs A_0 and B_0 being stimulated. Hence, $p(out)$ is the probability of the faulty NAND2 output being logic high. This formulation can be applied iteratively to compute circuit reliability. This shows that the PGM approach requires computation of the output distribution, given the input probability distributions and logic function (truth table) performed by the gate.

   The scalability problem affects as well this PGM methodology because matrices complexity increases exponentially with number of inputs and outputs. Thus, the disadvantage of this PGM methodology to compute reliability is that it is intractable for circuits in Nanotechnology as well as the PTM.

3. **Probabilistic Model Checking (PMC)**

   Bhaduri et al. [7] and Norman et al. [8] introduces the Probabilistic Model Checking (PMC) that is based on Discrete Time Markov Chains (DTMC) and the related transition probability matrices. PMC is a procedure for ascertaining whether a given probabilistic system satisfies probabilistic specifications. PMC has been used in [7] and [8] to evaluate the reliability of small circuits assuming that each gate fails independently from the other. Circuits are specified as DTMCs due to their suitability for modelling digital systems [9]. The reliability of a circuit is evaluated by computing the probability of reaching specific DTMC states, states that represent the correct Boolean values at the circuit outputs for a certain probability distribution at the inputs. Bhaduri et al. [7] even developped a certain probabilistic model checker called PRISM to analyze circuit DTMCs.

Figure 2.5 shows a DTMC of a logic NAND2 gate with inputs A_0 and B_0, and output out. The DTMC has 15 states and 22 transitions. The state variables are s, A_0, B_0 and out, where s is used to represent the level in the DTMC. Formally, each DTMC state is encoded as a quadruple (s, A_0, B_0, out), where $s \in$ 0,1,2,3 and A_0, B_0 and out are Boolean variables. The transitions in 2.5 have probability values attached to them. Let's take a simple example. The probability of 0.9 for transitioning from state 0 (encoded as (0,0,0,0)) to state 2 (encoded as (1,1,0,0)) implies that input A_0 has a 0.9 probability of beeing stimulated, i.e., beeing in logic "1" state. Similarly, a 0.01 probability of moving from state 6 (2,1,1,0) to state 14 (3,1,1,1) implies that the NAND2 gate can fail independently with a 0.01 probability.

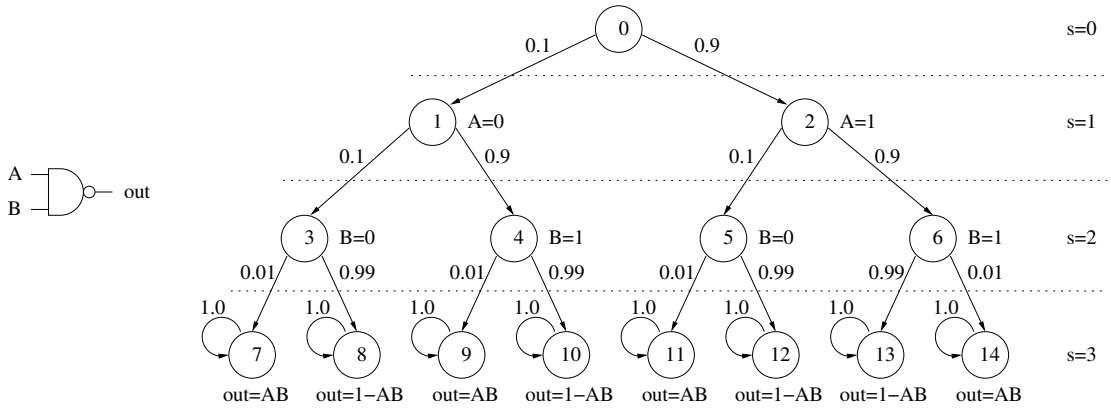

Figure 2.5: DTMC of a NAND2 gate

Matrix in 2.2 is a one step transition probability matrix of the DTMC of the Figure 2.5. Each row of the matrix represents the transition flow out of the corresponding state. For instance, the first row of the matrix shows that there is a probability of 0.1 for transitioning from DTMC state 0 to DTMC state 1, and a probability of 0.9 of moving from state 0 to state 2. Since the accumulative transition flow out of each state must be 1 in a DTMC, the rows of the matrix must sum to 1.

$$\begin{bmatrix}
0 & 0.1 & 0.9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.1 & 0.9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.1 & 0.9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.01 & 0.99 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.01 & 0.99 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.01 & 0.99 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.01 & 0.99 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} \tag{2.2}$$

4. **Probabilistic Binomial Reliability (PBR)**

   The work of Correia et al. [10] introduces a new model to compute logic circuit reliability, the Probabilistic Binomial Reliability model for Reliability analysis (PBR). The probabilistic binomial reliability analysis (PBR) method uses fault injection and functional simulation to determine an analytical model for the reliability of the circuit. Exploring the binomial probability distribution of multiple simultaneous faults, this reliability analysis method is capable of producing exact results or highly accurate ones for large circuits.

   The probabilistic binomial reliability model (PBR) for reliability analysis was developed as an analytical approach to model the reliability of fault-prone logic circuits, and implemented by means of fault-simulation or fault-injection in a circuit emulation environment.

   Consider that the gates of a circuit have a certain probability to fail, the reliability R of this circuit, seen as in a black box, can be determined as in 2.3, where $p(\vec{y} = correct|\vec{x}_i)$ represents a correct output considering a vector $\vec{x}_i$ and its probability of occurence $p(\vec{x}_i)$, and m the number of inputs in the circuit.

$$R = \sum_{i=0}^{2^m-1} p(\vec{y} = correct|\vec{x}_i)p(\vec{x}_i) \tag{2.3}$$

   As we consider all the combinations of possible inputs and fault-vectors of all G gates. In this case, the circuit reliability is as shown in the equation below 8.1, with the input vectors $\vec{x_i}$ and the fault-vectors $\vec{f_j}$

$$R = \sum_{j=0}^{2^G-1} \sum_{i=0}^{2^m-1} p(\vec{y} = correct|\vec{x}_i)p(\vec{x}_i)p(\vec{f_j}) \tag{2.4}$$

   This expression is even written into a more generalized and detailed equation with a logical masking expression, i.e, an expression representing the number of faults that gives correct output.

   The Probabilistic Binomial Reliability Model has proved to be exact in evaluating the reliability of combinational circuits. Despite the precision attained by this PBR method, the general application of the model is dependent on functional simulation, a task known to be time and resource consuming for practical circuits.

5. **Signal Probability Reliability (SPR)**

   The work of Franco et al. [11] introduces a new methodology to compute logic circuit reliability based on Signal Probabilities. The main advantage of the method is the simple algorithm for signal probabilities determination and propagation. The choice for a four-state signal probability representation is crucial for the propagation algorithm, and explicitly embeds the signal reliability information for any signal in the circuit.

   Franco defines a 2×2 matrix representation for any binary signal as can be seen in Figure 2.6, where $signal_0 = correct0$, $signal_1 = incorrect1$, $signal_2 = incorrect0$ and $signal_3 = correct1$.

$$SIGNAL_4 = \begin{bmatrix} signal_0 & signal_1 \\ signal_2 & signal_3 \end{bmatrix}$$

$$P_{2\times2}(signal) = \begin{bmatrix} P(signal = correct\ 0) & P(signal = incorrect\ 1) \\ P(signal = incorrect\ 0) & P(signal = correct\ 1) \end{bmatrix}$$

Figure 2.6: 2 x 2 probability signal matrix representation

   For each signal, another 2×2 matrix is also defined. This matrix gives the probabilities of occurence of these states as $P(signal = correct0)$, $P(signal = correct1)$, $P(signal = incorrect0)$ and $P(signal = incorrect1)$.

   The output signal probability matrix (S_4) of the NAND2 gate of Figure 8.1 can be determined by the joint input probabilities and the cell transfer function. The transfer function represent the fault-prone model of the logic function, i.e., the probability of correct operation and failure and its dependance with the reliability of the logic cell (gate).

   The joint input probabilities correspond to a tensor product (Kronecker product) of the gate input signal probabilities. The primary input signal probabilities consider fault-free signals. The propagation of the input signal probabilities through the NAND2 logic gate is computed by multiplying the gate joint input probability matrix (I_{gate}) by the transfer function of the gate, i.e., the Probabilistic Transfer Matrix of the NAND2 logic gate, as can be seen in equation 2.5.

$$P(S) = I_{gate} \times PTM_{NAND2} \tag{2.5}$$

   The output signal probability matrix (S_4) of the NAND2 gate is computed by regrouping the output probabilities in ($P(S)$) according to the logic function of

the gate, representated by its ITM. Expressions in B.1, 2.8, 2.9 and 2.7 represent the referred operation. The indexes $0, r$ and $1, r$ in these expressions correspond to the $column$, row indexes in the respective matrix.

$$p_0(signal) = \sum_{r:ITM(0,r)=1} P(signal)_{[0,r]} \tag{2.6}$$

$$p_1(signal) = \sum_{r:ITM(1,r)=0} P(signal)_{[1,r]} \tag{2.7}$$

$$p_2(signal) = \sum_{r:ITM(0,r)=0} P(signal)_{[0,r]} \tag{2.8}$$

$$p_3(signal) = \sum_{r:ITM(1,r)=1} P(signal)_{[1,r]} \tag{2.9}$$

Finally, the circuit reliability is the addition of the output probability of having a correct 0: p_0 and the output probability of having a correct 1: p_3. For multiple outputs, the reliability of the circuit can be obtained by the multiplication of the individual reliability of the output signals. The propagation procedure must be applied in a topological order, from the inputs of the circuit to the outputs, a single-pass approach of this methodology is used. The output signal probabilities represent the exact signal reliability of the circuit for the case where no signal correlations (reconvergent fanouts) exists. This approach of the SPR methodology is called Single-Pass because the reliability computation is done a unique time from the inputs to the outputs of the circuit.

A detailed example of the single-pass approach of reliability analysis based on signal probabilities is given in the Appendix A.

However, a lack of reliability accuracy of this single-pass approach above have been noticed due to the fact that it does not take into account reconvergent fanouts of the circuit. In order to overcome this problem, another algorithm had to be developed[12]. This algorithm that consists of propagating signal probabilities in multiple steps is called Multi-Pass SPR, in opposition to the original Single-Pass SPR described above. An example of this multi-pass approach of the signal probability-based reliability analysis is illustrated in 2.7, 2.8, 2.9 and 2.10 .

In fact, at each propagation process, only one signal probability is propagated. At the end of all iteration steps, all reliabilities calculated are accumulated as in 2.11 where J is the number of reconvergent fanouts in the circuit. Although this multi-pass methodology gives exact reliability results, its calculation complexity is enormous for large circuits with lots of fanouts and gates.

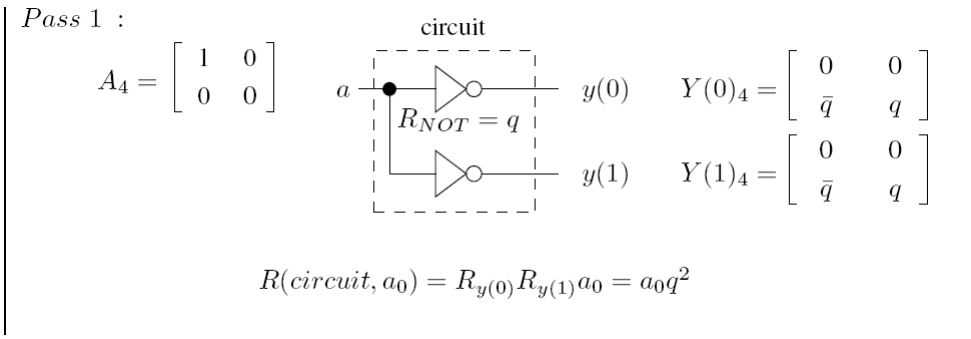

Figure 2.7: multi-pass: iteration 1

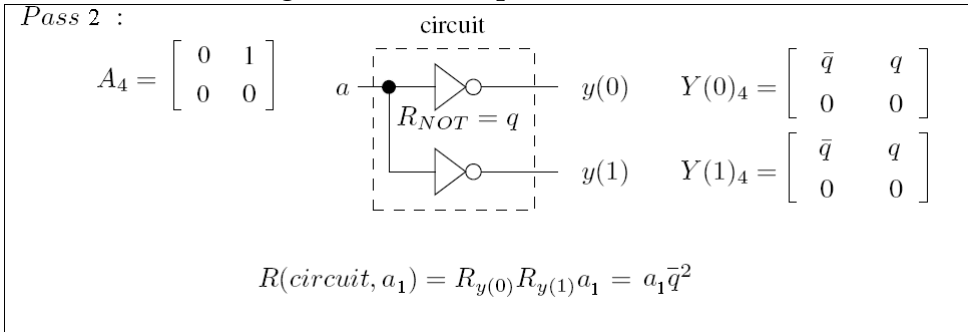

Figure 2.8: multi-pass: iteration 2

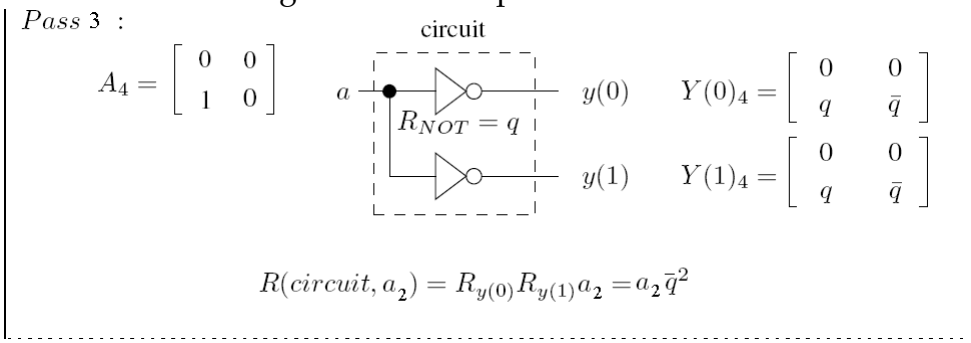

Figure 2.9: multi-pass: iteration 3

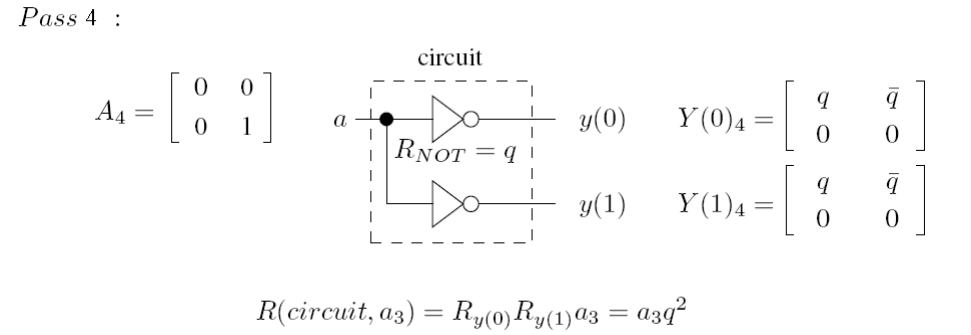

Figure 2.10: multi-pass: iteration 4

So the reliability $R_{circuit}$ using the example multi-pass approach of the SPR is as in 2.10

$$[!ht]R_{circuit} = \sum_{f=1}^{4^1} R_{circuit}(f) = a_0.q^2 + a_1.(1-q)^2 + a_2.(1-q)^2 + a_3.q^2 \quad (2.10)$$

$$[!ht]R_{circuit} = \sum_{f=1}^{4^J} R_{circuit}(f) \quad (2.11)$$

A detailed example of the multi-pass approach of reliability analysis based on signal probabilities is given in the Appendix B.

6. **Signal Probability Reliability using Weighted Averaging Algorithm (SPRWAA)**

   This method is derived from the Signal Probability Reliability (SPR) analysis methodology described above. Another methodology to get a good approximation of the circuit reliability value, near the exact circuit reliability of the multi-pass approach is the Weighted Averaging Algorithm (WAA) from the work of Krishnamurthy and Tollis [13]. This approach tries to correct the signal probabilities by computing the effect of individual input probabilities on these values, providing a root mean square deviation from the exact values. The time complexity of the algorithm is linear in the number of gates and input signals.

   The work of Ercolani et al. [14] proposes two heuristics for correcting signal correlation effects, the dynamic WAA (DWAA) and the correlation coefficients method. The DWAA applies the same corrections of the WAA approach but controls the reconvergent fanout signal probabilities instead of the input signal ones, and updates the signal probabilities at each iteration of the algorithm. The correlation coefficient method computes along with signal probabilities the conditional dependence of these signals. The complexity of this method is dependent on the circuit structure. The results of these approaches are better than the ones obtained with the WAA approach.

2.2.3 Problematic

As seen in the previous subsection concerning the state of the art above, a number of methodologies have been recently developed to evaluate circuit reliability. These methodologies that analyze the reliability independent of the technology used to fabricate nanoscale devises. Development of accurate reliability estimation methodologies is thus critical in analyzing the robustness of circuits designed from unreliable

nanoscale devices.

Evaluating the reliability of large circuits is very important since the realistic electronic systems are significantly large. The aforementioned reliability analysis do not scale well and, hence, are not efficient at evaluating sufficiently large circuits. For a circuit with m inputs and n outputs, the space and time complexity is generally proportional to 2^{m+n}, and there is a very high demand on memory and run-time for its reliability analysis.

Chapter 3

Principle of the proposed solution

As seen in the previous section, all reliability analysis methodologies are extremely huge in terms of memory usage and run-time, due to huge numerical computations needed. All methodologies have the same problem of scalability. The research work is to find new methodologies that can simplify the methodologies developed so far.

In order to make our work effective, we decided to focus our research on methodologies to simplify the SPR model, recently developed at the laboratory of the ENST. All these simplifications can be represented by different tradeoffs, i.e., compromises between three parameters: robustness (reliability accuracy), cost (memory usage for computing reliability) and speed (reliability computation run-time).

The real chalenge of our work is moreover to develop a new model having the advantages of the state of the art's models and modelize it as a tool for reliability analysis that can be furtherly integrated in the design flow of any logic circuit. This new model must be scalable unlike the previous reliability analysis models developed in the state of the art.

We propose in this section the main several approaches in the scope of our research for simplifying the SPR algorithm for reliability analysis, and a new methodology that can be the key feature for our new reliability analysis model and its tool.

3.1 SPR simplifications

3.1.1 Vector SPR approach

Using the signal probability-based algorithm for reliability analysis, the effect of reconvergent fanouts can be taken into consideration by the calculation of the reliability of the circuit in multiple propagation steps. In fact, in the multi-pass algorithm, in order to get the exact value of the circuit reliability, we were considering the propagation of only one signal probability state of a reconvergent fanout at each iteration step, before accumulating all the results together. Depending on the number

of reconvergent fanout J in the circuit, the number of calculation was equal to 4^J , which is extremely huge for large circuits with lots of signal correlations. Figure 3.1 shows a demonstration of the situation for 2 or 3 fanouts.

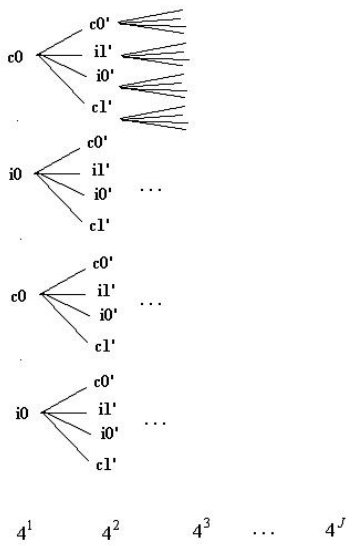

Figure 3.1: Calculation Number for a circuit having fanouts in a Multi-pass Approach

The final value for the circuit reliability is in fact the summation of all these reliabilities calculated at each step of iteration as can be seen in 3.1.

$$R_{circuit} = \sum_{x=1}^{4^J} R_{circuit}(x) \tag{3.1}$$

Otherwise, having more than one signal probability state of a reconvergent fanout in one iteration, would make a compromise in terms of complexity of calculation between the multi-pass algorithm, which consider the propagation of only one signal probability state at a time, and the single-pass algorithm, which consider the propagation of the four signal probability states at a time.

Consider a multi-pass algorithm propagating two states of signal probability at each iteration, i.e. a vector of signal probabilities. Three choices of the signal probabilities can be done according to the vectors of signal probabilities chosen horizontally, vertically or diagonally as shown in Figure 3.2.

In all three cases, we have a reduction in complexity of calculation. Let's take an example with the vectors of signal probabilities chosen horizontally as in Figure 3.3.

For all three cases, the final value of the circuit reliability is a summation of the 2^J reliabilities calculated at each iteration process. The circuit reliability can be then written as in 3.2.

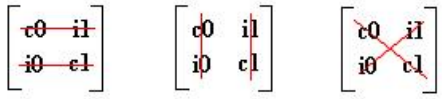

Figure 3.2: Horizontal, Vertical and Diagonal Choices of Signal Probability Vectors

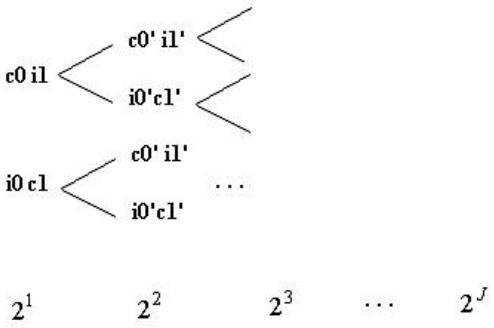

Figure 3.3: Calculation Number for the Horizontal choice of Signal Probability Vectors

$$R_{circuit} = \sum_{x=1}^{2^J} R_{circuit}(x) \tag{3.2}$$

Let's compare now the number of calculation N done with the multi-pass Algorithm considering one signal probability propagation at a time $MP1$ and with the multi-pass algorithm considering two signal probabilities propagation at a time MP2, as in 3.3.

$$\frac{MP1}{MP2} = \frac{4^J}{2^J} = 2^J \geq 2 \;\; if \;\; J \geq 1 \tag{3.3}$$

Thus, the complexity of calculation is reduced to more than the half considering two signal probability states at each iteration process. But numerical analysis are in need to study the deviation of the circuit reliability from the exact value.

3.1.2 Fanouts neglection

Another interesting solution to make a tradeoff between the complexity of calculation and the accuracy of the circuit reliability value, i.e., its deviation from the exact circuit reliability value obtained if using the multi-pass approach of the SPR methodology is the fanouts neglection. Fanouts neglection approach considers that some of the circuits reconvergent fanouts can be neglected if their effects are huge in terms of number of calculation.

In fact, we have seen in the previous subsection "Vector SPR Approach" that the multipass approach, considering the effect of all reconvergent fanouts of the circuit J needs 4^J iteration steps to obtain the exact circuit reliability value as can be seen in the equation of 3.1. But, these 4^J reliability calculations are not the same in terms of complexity of calculations. In fact, the more the reconvergent fanouts are located near the end levels of the considered circuit, the more the complexity of calculation is negligible. The level corresponding to the nearest reconvergent fanout to the inputs of the circuit, represent the set point level where all calculations are done 4^J times. The more we can move away this nearest reconvergent fanout neglecting it, the more the set point level is moved away, and the complexity of calculation is reduced.

A study of the effect of this neglection on the accuracy of the circuit reliability is in need to be made in order to analyze the compromise between the reduction of the complexity calculation and the accuracy of the circuit reliability value.

3.1.3 Fanout signal matrices simplifications

Another solution to help compromising complexity of computation and circuit reliability deviation from the exact value, is to simplify the matrices corresponding to the signals of the circuit where fanouts exist.

In fact, in the Appendix B relating to the multi-pass approach of the signal probability-based algorithm, the example circuit shows that having $P(signal = incorrect0) = 0$ and $P(signal = incorrect1) = 0$ enables us to make only two iterations instead of four iterations due to the fact that these zeros in the fanout matrix gives null results anyway.

Finding a method to neglect the incorrect probabilities of the 2×2 signal probability matrix of the fanout signal node, would decrease a lot the complexity of computation of the circuit reliability.

3.2 Scalable PTM Approach

As seen in the state of the art, scalability has been a worry for large circuits. In fact, most of the time, monolithic models of large nanoscale circuits (realistic electronic systems), which consist of millions to billions of devices, have such huge memory usage that their reliability analysis become intractable.

An algorithm of scalability for signal probability-based methodologies can be then used in order to reduce complexity of calculation and allow reliability analysis of large nanoscale circuits with less significant memory usage, or otherwise according to designer's capacity of memory and designer's time requirements for evaluating his circuit reliability.

Bhaduri et al. [7] presents a scalability technique for the case of a Probabilistic Model Checking (PMC) algorithm of circuit reliability analysis. This method consists of iteratively building a DTMC (Discrete Time Markov Chains) model for each gate in a level (in order to build its Transition Probability Matrix), and freeing up memory used to represent this model once analyzing all the gates outputs in the corresponding level, and doing the same three steps for the following level, until finishing analyzing the whole circuit.

The algorithm of scalability consists of:

i) topologically partitioning the full circuit by doing a levelized-list of all the circuits' gates
ii) iteratively build and analyze DTMC model of each partition to compute gates' outputs
iii) propagating these outputs as inputs for the following partition and finally clearing matrices done in the preceding partition.

An algorithm of scalability for the PTM model would be a key feature for nanoscale circuits' reliability analysis, having the problem of beeing intractable.

An example of scalable PTM tool in Figure 3.4 is composed of:

- **Initialization**: reads the circuit netlist, and the components reliabilities
- **circuit description matrix**: it is a $m \times n$ matrix, where m is the maximum number of gates per level, and n the number of levels. This matrix describe the components composition of the whole circuit level by level.
- **scalable algorithm**: it consist of iteratively get the **input level matrix**, then calculate the **ITM** (using the ITM components library) and **PTM level matrices**, and finally use these matrices to compute the **propagation matrix** and the **input level matrix** of the following level. Note that the respective matrices (input, ITM, PTM, propagation) of level i are written onto the respective matrices (input, ITM, PTM, propagation) of level $i - 1$. This approach of clearing the memory reduces a lot the memory storage during the reliability computation.
- **circuit reliability**: Finally, for only the last level, the reliability is computed and saved in a .dat file. This reliability represents the whole circuit reliability.

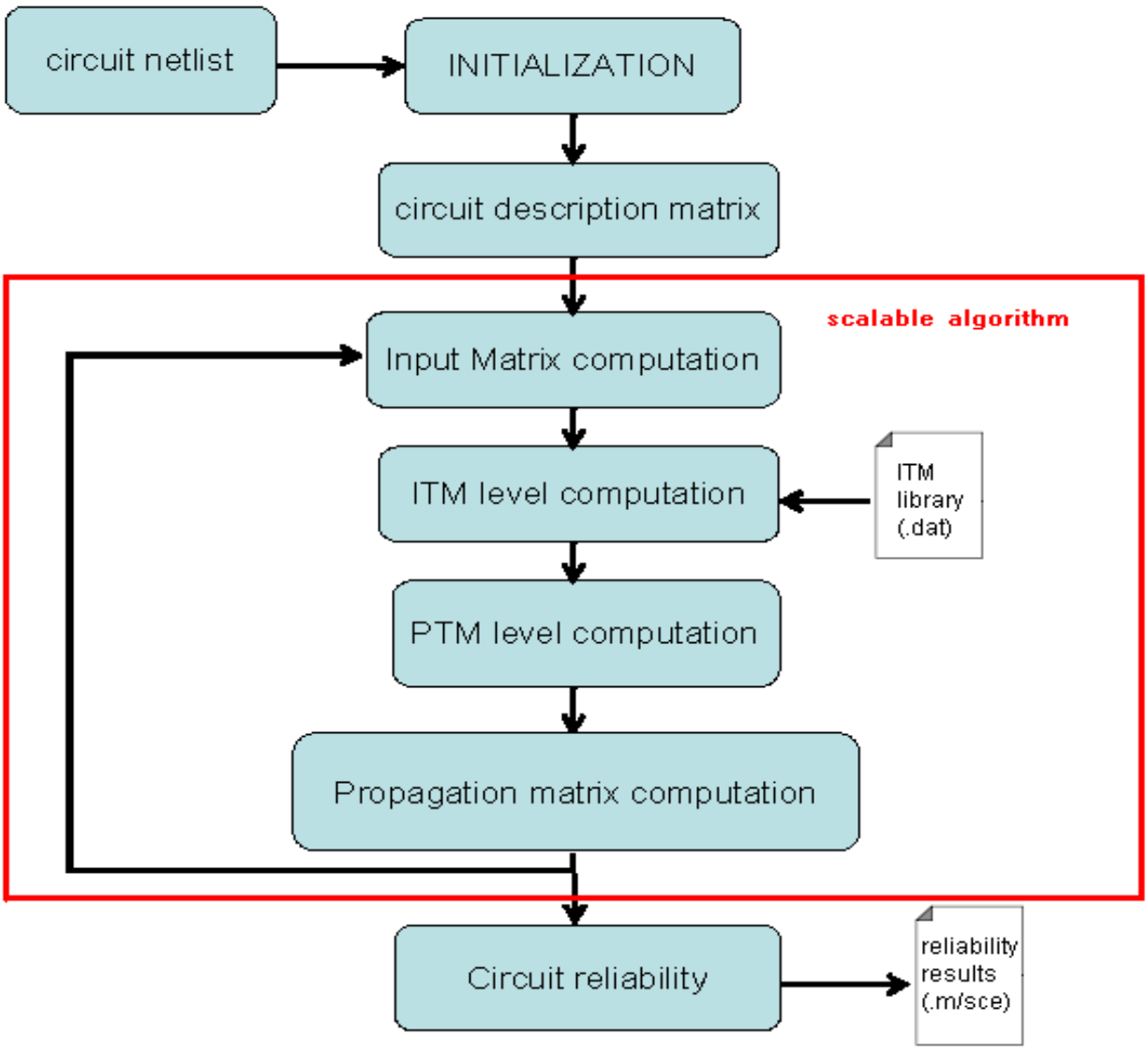

Figure 3.4: example of a scalable PTM tool

Chapter 4

Tasks Identification

Having seen the proposed solutions to resolve the problematic, this section presents and identifies the main tasks related to SPR simplifications and our new model and tool based on scalable PTM for reliability analysis.

4.1 Implementation and analysis of SPR proposed simplifications

As seen in the previous section, SPR reliability analysis model can be simplified with lots of tradeoffs between complexity and accuracy. A numerical analysis can be then done to the SPR algorithm in order to reach approximate exact results and compare it with the classical SPR results.

Concerning the Vector SPR approach of the SPR algorithm, having already proved its reduction in terms of computation complexity in the section "Vector SPR Approach", some numerical analysis should be done in order to make a study of the deviation of the circuit reliability from the exact values. Using matrices computation in our numerical analysis, as multiplications, additions, kronecker products, etc... Matlab or Scilab would with no doubt be the available tools for such jobs.

Concerning the fanouts neglection and fanout signals matrices simplification, numerical analysis would also be done in order to open new ways and tracks for reliability analysis tradeoffs.

4.2 Implementation of PTM improvements

As seen in the previous section, a scalable PTM approach of the PTM model for computing circuit reliability can be very interesting to reduce a lot the memory usage for these computations. A tradeoff between computation run-time and computation complexity is then created.

4.2.1 Scalable PTM Model Matlab/Scilab Implementation

As for the SPR simplifications, Matlab or Scilab would be the best numerical and programmable tool to deal with all the matrices calculations. A programmable code written in Matlab or Scilab, can be done in order to compute logic circuit reliability. First of all, this code would be manually descriptible, i.e., the circuit to be tested would be manually created by the designer itself.

4.2.2 Evaluation through Fault-Tolerant Library

Using the circuit netlist interface, a library of fault-tolerant arithmetic functions can be then used as case studies for our scalable PTM reliability analysis tools. In fact, to evaluate the functionality of some fault-tolerant approaches a library of arithmetic circuits, adders and multipliers can be used. This library already exists and is used in lots of the state of the art papers for testing reliability analysis models.

The library is composed of standard adders and a booth multiplier. The traditional adder topologies, i.e., ripple-carry, carry-select and carry-lookahead, and the booth multiplier, represent well known tradeoffs between area, speed and power and a further characterization of these circuits by means of signal reliability is interesting for a more detailed characterization of the respective design space.

The circuits in the library are described in the VHDL language, with parameterized data widths, and targeted to standard cell synthesis. This library can then extract netlists that can be used to compute circuit reliability using our scalable PTM model.

Chapter 5

Procedure Definition

The main objectives of the current work are the development of new reliability analysis method and testing this method on logic circuits. The focus procedure is then to:

1. validate our simplifications approaches of the SPR methodology:

   Vector SPR

   Fanout Neglections

   Fanout Matrices Neglection

2. validate our proposed scalable reliability analysis tool based on PTM

The validation of the SPR simplification approaches is mainly a study of the tradeoffs between accuracy and processing time that will be evaluated considering the relaxation procedures discussed on the current work, in order to allow the application of the analysis to practical systems.

The validation of the reliability analysis tools mainly concerns their performance and practical limits from the point of view of the implemented simplifications, since the proposed methodologies model the exact behavior of the signal reliability metric.

Thus, to prove and validate our approaches concerning the SPR algorithm, and our new model for reliability analysis, several results based on graphs and logic circuit tests must be done. These results would be the key of a comparison between our methodology and previous methodologies proposed in the state of the art.

The main tradeoffs of comparison for SPR simplifications methodologies will be:

- tradeoff between complexity of calculation and logic circuit reliability value accuracy for the Vector SPR approach.
- tradeoff between complexity of calculation and logic circuit reliability value accuracy for the Fanout Neglection and Fanout Matrices Simplifications approaches.

Chapter 6

Tasks Planning

Table 6.1 resumes the essential tasks planning, and in general their delimited deadlines to respect for the following weeks to come.

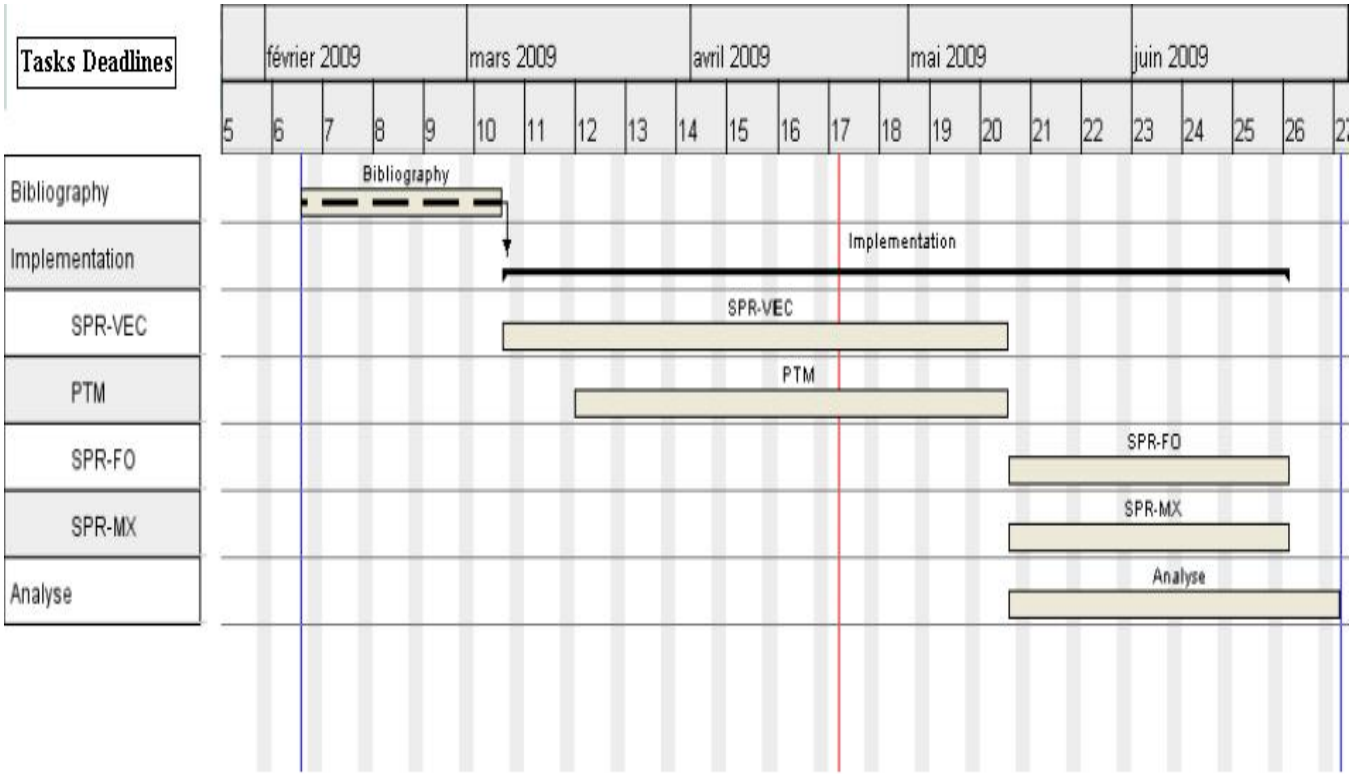

Figure 6.1: Tasks Planning & Deadlines

Chapter 7

Results: SPR compromises Study

7.1 Introduction

The signal correlation problem has been studied for a long time [15, 14] and many tradeoffs have been proposed to deal with the exponential complexity of probability correlations.

The effects of reconvergent fanouts are taken into account by the calculation of the reliability of the circuit in multiple propagation steps. In this multi-pass propagation process, only one signal probability state (of a reconvergent fanout) is propagated at a time and the resulting output reliability is accumulated along the multiple iterations.

As we have seen in the precedent sections, at the end of the multi-pass algorithm, the exact reliability value has been determined, as in 7.1, where F is the number of reconvergent fanout nodes and f represents the iteration f of the algorithm.

$$R_{circuit} = \sum_{f=1}^{4^F} R_{circuit}(f) \tag{7.1}$$

Despite the expected complexity, dependent on the number of reconvergent fanouts, many optimizations are possible. We propose in this work 3 possible compromises that reduces a lot the complexity of computation and run-time as seen in the previous sections. The 3 compromises are: SPR fanout neglection, SPR fanout matrix simplification and the SPR vector. Despite the memory and time reduction of these optimizations, let's see how each one of them deviates from the exact circuit reliability value obtained if we had to use the multi-pass algorithm of the SPR.

7.2 SPR Compromises

7.2.1 SPR fanout neglection

The accuracy of the SPR is dependent on the number of reconvergent signals that are taken into account by the multi-pass algorithm and the topology of the corresponding circuit. First of all, a process of determination of all the correlation among the signals in a circuit is done, then a process of sorting the fanout signals according to these correlations is made. Finally, considering this fanout list, the reliability analysis tool can execute the multi-pass algorithm based on a subset of the fanout signals.

Let's consider the circuit C17 from the ISCAS'85 benchmark, as shown in Fig.7.1.

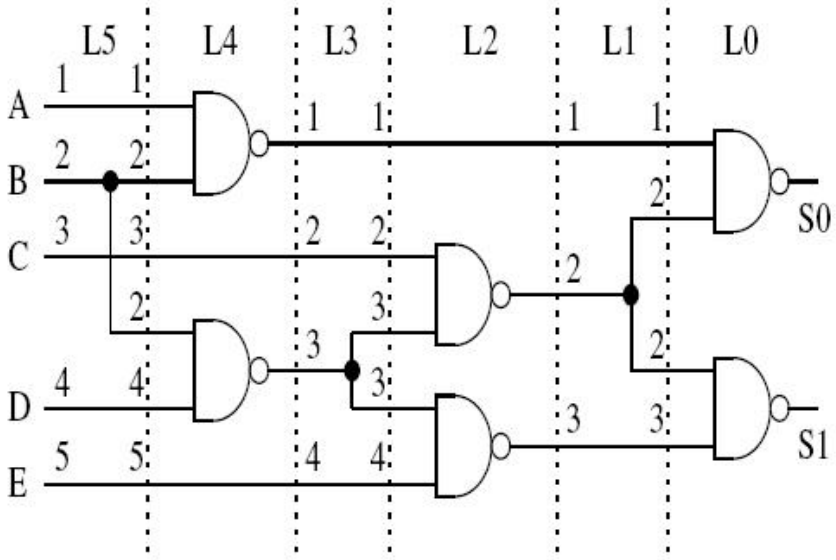

Figure 7.1: circuit C17 from ISCAS'85 benchmark

The C17 benchmark presents 3 fanout signals, in the node B of level L_5 and in the nodes of levels L_1 and L_3.

To determine the exact value of reliability of the circuit, all of the fanout signals must be taken into account, but approximate results can be obtained with a reduced number of fanouts. Table 7.1 below presents the reliability results when a different number of fanouts is taken into account.

The error value presented is the mean error in the reliability of the circuit concerning a cell reliability range $0 \leq R_{cell} \leq 1$. The error value is relative to reliability values obtained with the exact multi-pass algorithm of the SPR methodology for reliability

fanouts	error(%)
3	0
2	0.31
1	1.32
0	4.41

Table 7.1: C17 reliability error

analysis.

As can be seen in the table 7.1, the impact of the fanout signals is dependent of its logic distance to the output signals and its fanout cone, and the contribution of some fanouts can be ignored with no significant effect on the overall reliability.

In fact, the more the fanouts of the circuit are neglected, the more their contribution of reliability accuracy reduction is considerable. For instance, with 2 fanouts of the circuit C17, i.e., 1 fanout neglected, the reliability error is 0.31% while with 1 fanout of the circuit C17, the error is 1.32%.

Moreover, the more the fanout is distant from the inputs, the more its contribution to increase the deviation from the exact circuit reliability value is bigger. To take a numerical example, table 7.2 shows the situation of neglecting only one fanout but topologically in distance from the inputs to the outputs.

fanout level	error(%)
5	0.31
3	0.47
1	0.70

Table 7.2: C17 reliability error

Let's consider a clocked D Latch, as shown in Fig.7.2 having 2 fanouts.

Table 7.3 shows the same results from the analysis of an D Latch, with cell reliability in the range $0 \leq R_{cell} \leq 1$.

fanouts	error(%)
2	0
1	3.88
0	11.63

Table 7.3: D Latch reliability error

The graph of Fig.7.3 shows the differences in reliability values using the exact multi-pass algorithm of the SPR, the single-pass and the fanout neglection optimization.

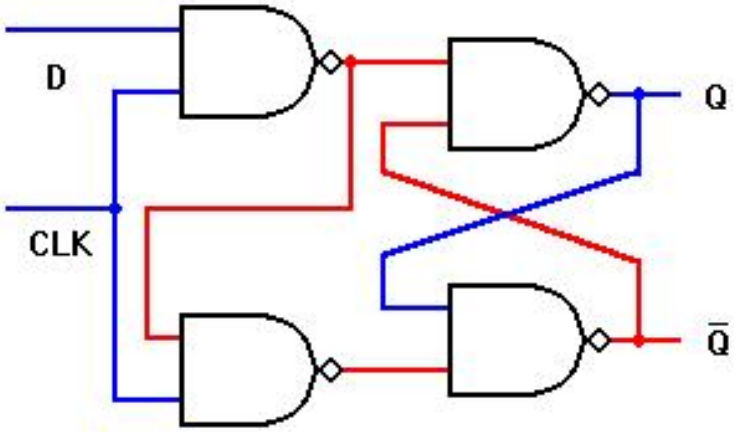

Figure 7.2: clocked D Latch

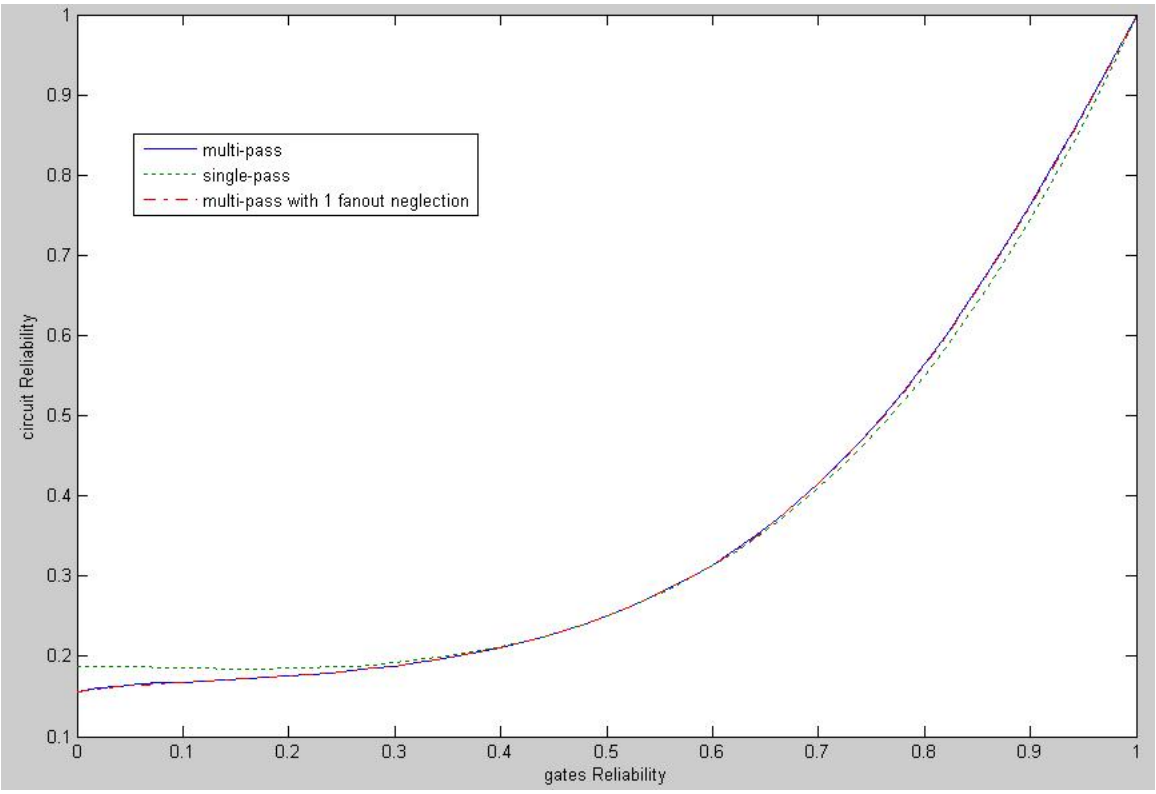

Figure 7.3: D Latch with fanout neglection

We note a maximum deviation of 16.58 % of the single-pass from the accurate multi-pass algorithm, while a maximum devialtion of % for the fanout neglection case as seen in Fig.7.4.

Let's now explore the same situation with several architectures of adders beginning with the most simple adder: the Half-Adder, as shown in Fig.7.5 having 2 fanouts.

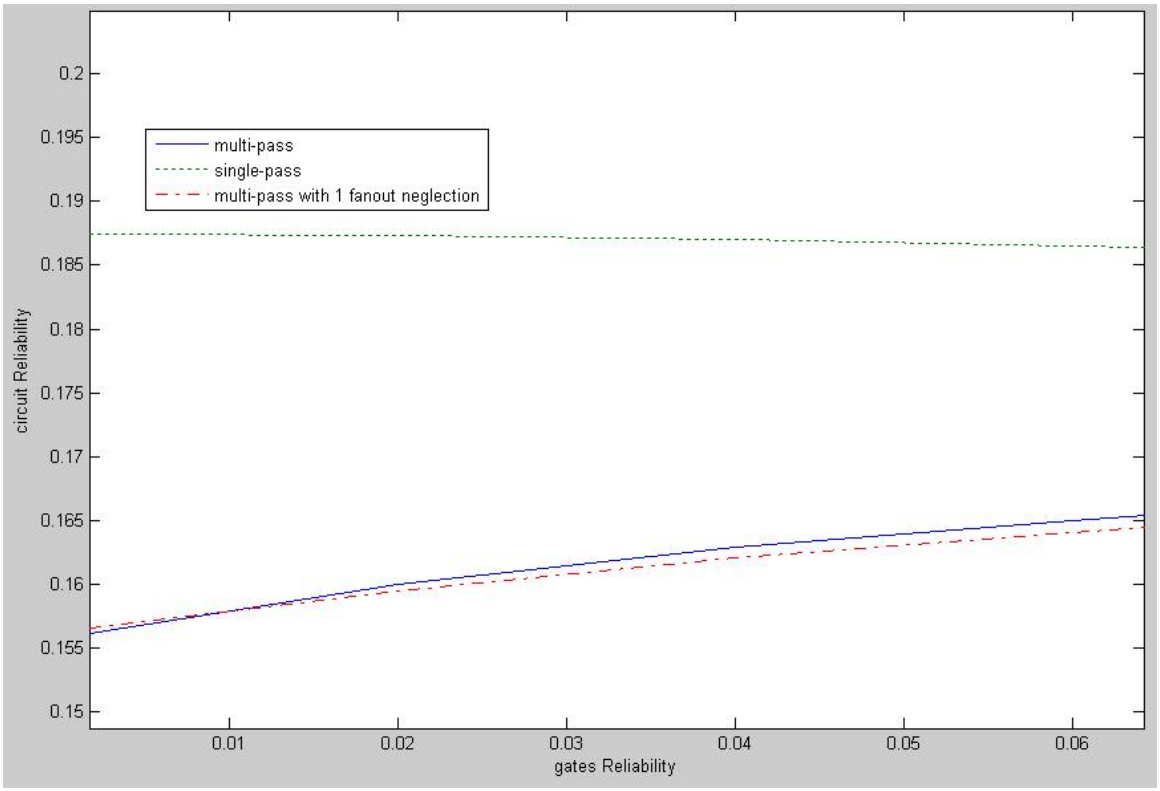

Figure 7.4: maximum deviation for the D Latch with fanout neglection

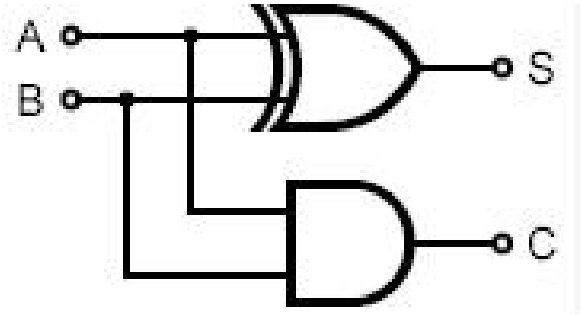

Figure 7.5: Half-Adder

Table 7.4 shows the same results from the analysis of a half-adder, with cell reliability in the range $0 \leq R_{cell} \leq 1$.

Let's now see the full-adder, as shown in Fig.7.6 having 4 fanouts.

Table 7.5 shows the same results from the analysis of a full-adder, with cell reliability in the range $0 \leq R_{cell} \leq 1$.

fanouts	error(%)
2	0
1	0.14
0	1.13

Table 7.4: Half-Adder reliability error

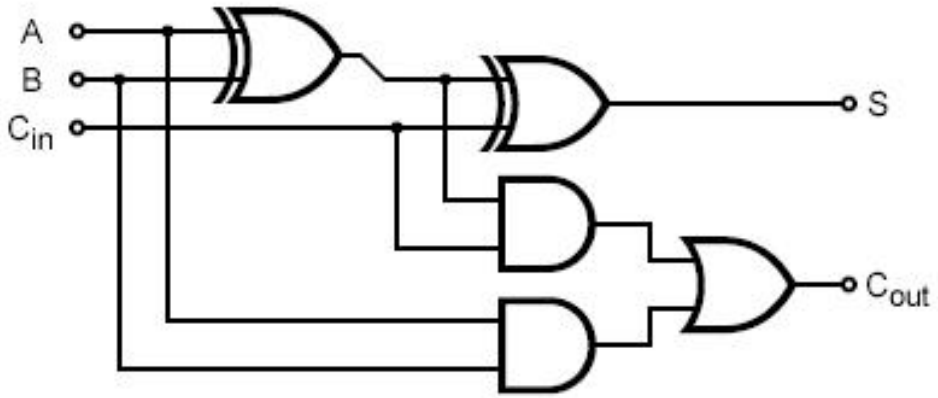

Figure 7.6: Full-Adder

fanouts	error(%)
4	0
3	0.05
2	0.26
1	1.29
0	2.11

Table 7.5: Full-Adder reliability error

Let's now explore the situation with the ripple-carry adder, the carry-select adder and the carry-skip adder having the same function but different number of fanouts.

Table 7.6 shows the results from the analysis for these architecture of adders, with cell reliability in the range $0 \leq R_{cell} \leq 1$.

As can be seen in all the different architectures of circuits and adders, the impact of the fanout signals is dependent of its logic distance to the output signals and its fanout cone, and the contribution of some fanouts can be ignored with no significant effect on the overall reliability.

The graph below in Fig.7.8 shows the percentage of error curves in function of the number of fanouts neglected (FN) for different circuit architectures.

fanouts	ripple carry error(%)	carry select error(%)	carry skip error(%)
12	X	X	0
11	X	X	0.11
10	X	0	0.22
9	X	0.23	0.31
8	0	0.26	0.75
7	0.03	0.37	1.04
6	0.17	1.01	1.28
5	0.20	1.52	1.89
4	0.21	1.74	3.72
3	0.77	1.86	3.90
2	1.09	3.11	4.02
1	3.18	3.44	4.12
0	4.50	3.86	5.93

Table 7.6: Carry Ripple, Select and Skip Adders reliability errors

In fact, the more the fanouts of the circuit are neglected, the more their contribution of reliability accuracy reduction is considerable.

The multi-pass algorithm still presents many possibilities for optimization and we present in the following section the SPR optimization consisting of simplifying fanout matrices. Besides that, another signal correlation procedures will be studied to determine the best compromise between accuracy and performance for the proposed algorithm.

7.2.2 SPR fanout matrix simplification

The other proposed solution to help compromising complexity of computation and circuit reliability deviation from the exact value, is the simplification of the matrices corresponding to the signals of the circuit where fanouts exist.

In fact, we have seen in the Appendix B relating to the multi-pass approach of the signal probability-based algorithm, showing the example circuit that having $P(signal = incorrect0) = 0$ and $P(signal = incorrect1) = 0$ enables us to make only two iterations instead of four iterations due to the fact that these zeros in the fanout matrix gives null results anyway.

Thus, neglecting the incorrect probabilities of the 2×2 signal probability matrix of the fanout signal node, would help decreasing a lot the complexity of computation of the circuit reliability.

Let's see what happens to the circuit C17 benchmark if we apply the corresponding neglection one by one of the matrices of the signals having fanouts. For instance, for

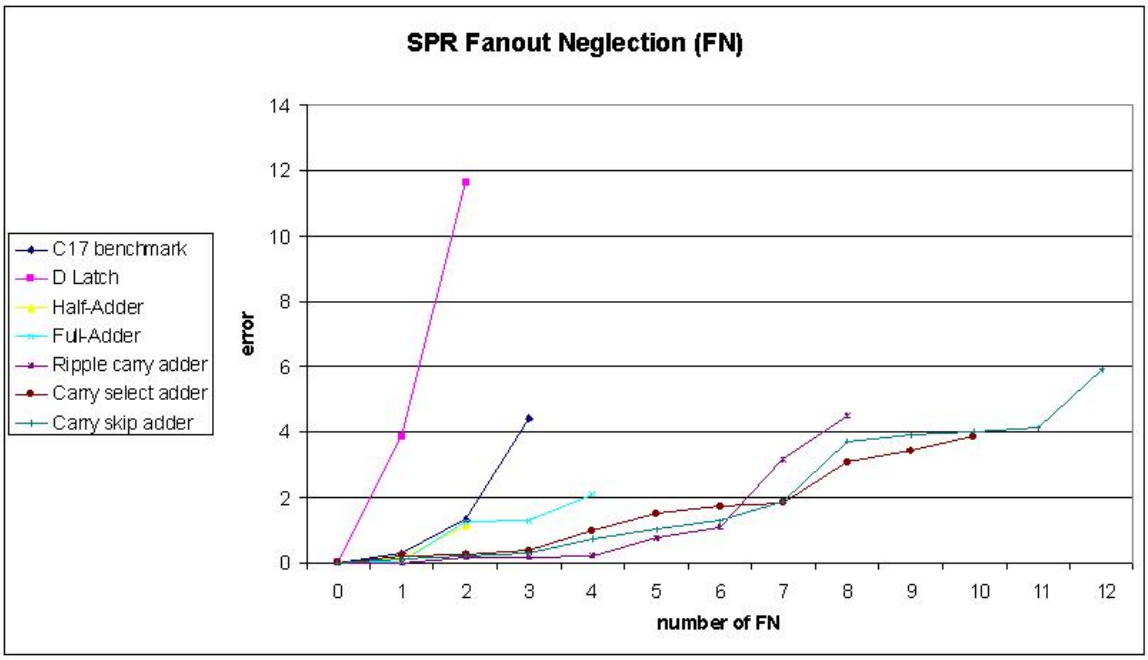

Figure 7.7: SPR Fanout Neglection percentage of error curves

the C17, we begin topologically by neglection the matrices of the signals in distance from the inputs until the outputs of the circuit.

Table 7.7 shows the results from the analysis of the multi-pass algorithm using the fanout matrices simplifications (FMS) for the C17, with cell reliability in the range $0 \leq R_{cell} \leq 1$. FMS represented in the table means the number of topological fanout matrices simplified, i.e., having its incorrect probabilities (the probability of $incorrect1$ and the probability of $incorrect0$) to null.

FMS	error(%)
0	0
1	4.76
2	7.43

Table 7.7: C17 reliability error

We can directly note that just simplifying 1 of the 3 fanouts matrices gives more reliability accuracy error than the error obtained using single-pass itself. In fact, for the C17 circuit, we obtain a circuit reliability accuracy error of 4.41%, while for the first fanout matrix simplified, we obtain an error of 4.76%, which is bigger than the multi-pass case. Moreover, adding to this the simplification of the second fanout matrix approximately doubles the error.

Let's try other architectures of circuits.

Table 7.8 shows the results from the analysis of the multi-pass algorithm using the fanout matrices simplifications (FMS) for the D Latch, with cell reliability in the range $0 \leq R_{cell} \leq 1$.

FMS	error(%)
0	0
1	9.45
2	23.23

Table 7.8: D Latch reliability error

We can see again here, for the example of the D Latch that the simplification of the second fanout matrix deviates more from the multi-pass exact value than the single-pass algorithm. In fact, for the C17 circuit, we obtain a circuit reliability accuracy error of 11.63%, while for the first fanout matrix simplified, we obtain an error of 23.23%, which is double the single-pass circuit reliability error obtained.

Let's see what happens using different architectures of adders from different sizes.

Table 7.9 shows the results from the analysis of the multi-pass algorithm using the fanout matrices simplifications (FMS) for different architectures of adders (half-adder, full-adder, ripple carry adder, carry select adder and carry skip adder), with cell reliability in the range $0 \leq R_{cell} \leq 1$.

FMS	half-adder error(%)	full-adder error(%)	ripple-carry adder error(%)	carry-select adder error(%)	carry-skip adder error(%)
0	0	0	0	0	0
1	4.07	2.90	6.18	5.32	13.12
2	6.27	4.55	19.48	14.30	26.84

Table 7.9: adders reliability error

We can see again here, for the all the different architectures of adders presented, that simplifying one or more fanout matrices gives always errors bigger than the error obtained if we use a single-pass algorithm.

The graph below in Fig.**??** shows the percentage of error curves in function of the number of fanouts matrix simplified (FMS) for different circuit architectures.

For all circuit, the very high error obtained using the FMS optimization method is due to the fact that neglecting the incorrect probabilities of the 2×2 signal probability matrix (the probability of $incorrect1$ and the probability of $incorrect0$) of the fanout

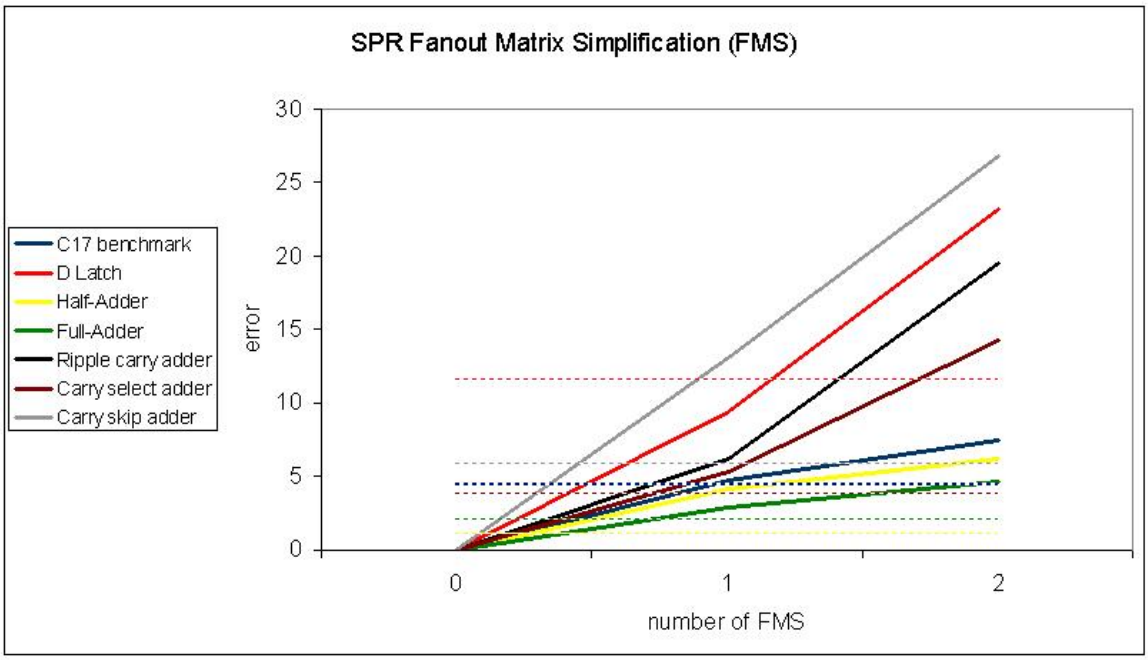

Figure 7.8: SPR Fanout Matrix Simplification percentage of error curves

signal node, help decreasing a lot the complexity of computation of the circuit reliability, but gives incorrect values of reliability.

We can conclude from this that it is not possible to neglect the values of the incorrect signal probabilities, because it gives less accurate reliability results than the simple single-pass itself. Thus, neglecting incorrect probabilities of fanout matrices cannot be considered a way of optimization of the SPR model in terms of accuracy of reliability value.

Thus, the compromise of simplifying fanout matrices can not be taken into consideration in the scope of our study of SPR compromises. Let's make a study of the last SPR compromise proposed: the SPR vector optimization.

7.2.3 SPR vector

We have seen that having more than one signal probability state of a reconvergent fanout in one iteration, would make a compromise in terms of complexity of calculation between the multi-pass algorithm, which consider the propagation of only one signal probability state at a time, and the single-pass algorithm, which consider the propagation of the four signal probability states at a time.

The SPR vector simpy consider a multi-pass algorithm propagating two states of signal probability at each iteration, i.e. a vector of signal probabilities, instead of a one

by one state of probability as done for the classical multi-pass algorithm of the SPR model.

We have seen in the sections above that the complexity of calculation is reduced to more than the half considering two signal probability states at each iteration process, but numerical analysis help us to make a study of the deviation of the circuit reliability from the exact value.

Let's consider the circuit C17 from the ISCAS'85 benchmark of Fig.7.1.

Table 7.10 below presents the reliability results using the SPR vector algorithm.

SPR single-pass error(%)	4.41
SPR multi-pass error(%)	0
SPR vector error(%)	2.07

Table 7.10: C17 reliability error

The error value presented is again the mean error in the reliability of the circuit concerning a cell reliability range $0 \leq R_{cell} \leq 1$. The error value is relative to reliability values obtained with the exact multi-pass algorithm of the SPR methodology for reliability analysis.

We can see in the table 7.10 that the error for the SPR vector is reduced to half the error obtained using the single-pass algorithm of the SPR methodology.

Let's consider now the clocked D Latch of the Fig.7.2.

Table 7.11 shows the same results from the analysis of an D Latch, with cell reliability in the range $0 \leq R_{cell} \leq 1$.

SPR single-pass error(%)	11.63
SPR multi-pass error(%)	0
SPR vector error(%)	2.14

Table 7.11: D Latch reliability error

We can see in the table 7.11 that the error for the SPR vector is extremely reduced to more than $1/5$ the error obtained using the single-pass algorithm of the SPR

methodology, which is an excellent approximate result near the exact multi-pass algorithm of the SPR model.

Let's now explore the same situation with several architectures of adders beginning with the most simple adder: the half-adder of Fig.7.5.

Table 7.12 shows the same results from the analysis of a half-adder, with cell reliability in the range $0 \leq R_{cell} \leq 1$.

SPR single-pass error(%)	1.13
SPR multi-pass error(%)	0
SPR vector error(%)	0.86

Table 7.12: Half-Adder reliability error

We can see in the table 7.12 that the error for the SPR vector is reduced to the single-pass algorithm.

Let's now see the full-adder of the Fig.7.6.

Table 7.13 shows the same results from the analysis of a full-adder, with cell reliability in the range $0 \leq R_{cell} \leq 1$.

SPR single-pass error(%)	2.11
SPR multi-pass error(%)	0
SPR vector error(%)	1.12

Table 7.13: Full-Adder reliability error

We can see in the table 7.13 that the error for the SPR vector is reduced to half the error obtained using the single-pass algorithm of the SPR methodology.

Now let's explore the situation with the ripple-carry adder, the carry-select adder and the carry-skip adder having the same function.

Table 7.14 shows the same results from the analysis of a half-adder, with cell reliability in the range $0 \leq R_{cell} \leq 1$.

adder architectures	ripple adder	select adder	skip adder
SPR single-pass error(%)	4.5	3.86	5.93
SPR multi-pass error(%)	0	0	0
SPR vector error(%)	2.98	1.77	2.19

Table 7.14: Carry Ripple, Select and Skip Adders reliability errors

We can see again in the table 7.14 that the error for the SPR vector is reduced to approximatively half the error obtained using the single-pass algorithm of the SPR methodology.

The figure Fig.7.9 shows the percentage of error using the SPR vector approach for different circuit architectures.

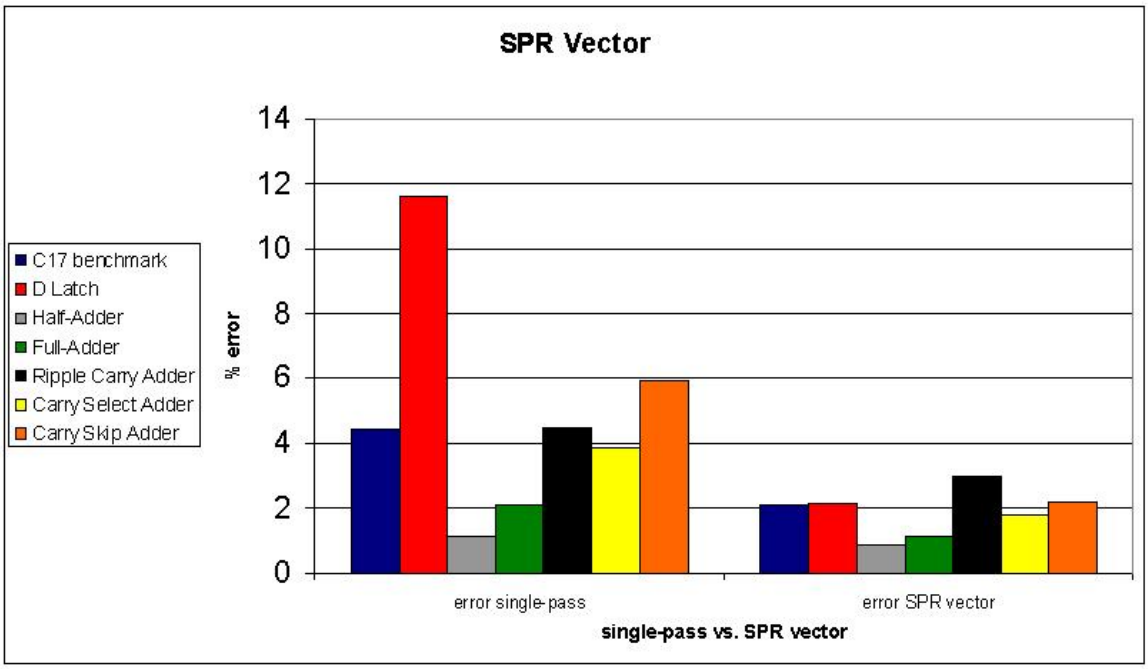

Figure 7.9: SPR vector percentage of error

In conclusion, the optimization done using the SPR vector algorithm of the SPR model, not only reduce the computation complexity and run-time, but also reduces the circuit reliability error in comparison to the single-pass algorithm of the SPR.

7.3 Statistical analysis of the SPR compromises

Having validate 2 of the 3 compromises, let's now statistically compare between the SPR fanout neglection and the SPR vector. In fact, we do not take in account the third compromise of SPR fanout matrix simplification as its deviation errors from the exact multi-pass algorithm are bigger than the simple single-pass errors.

The graph of Fig.7.10 shows the circuit reliability versus the gates reliability for the different circuits we have used in our study: C17 benchmark, clocked D Latch, half-adder, full-adder, ripple carry adder, carry select adder and carry skip adder.

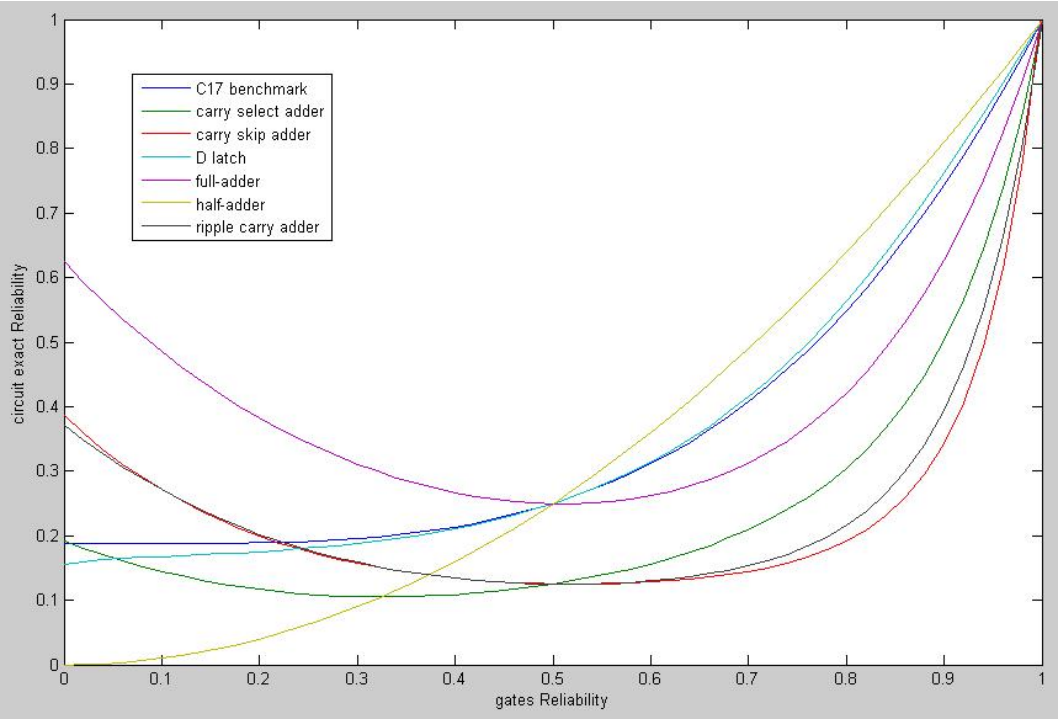

Figure 7.10: Circuits reliability vs. gates reliability

Fig.7.11, Fig.7.12, Fig.7.13, Fig.7.14, Fig.7.15, Fig.7.16 and Fig.7.17 shows the statistical analysis of the 2 validated compromises for the SPR methodology for reliability analysis.

In all the presented graphs above, for each circuit, we can see that:

1. each circuit has its own characteristical reliability graph, which is the corresponding circuit reliability versus the circuit's gates reliability.
2. Either using the SPR vector optimization or the SPR fanout neglection optimization, it has always circuit reliability accuracy errors smaller than the single-pass circuit reliability accuracy errors.

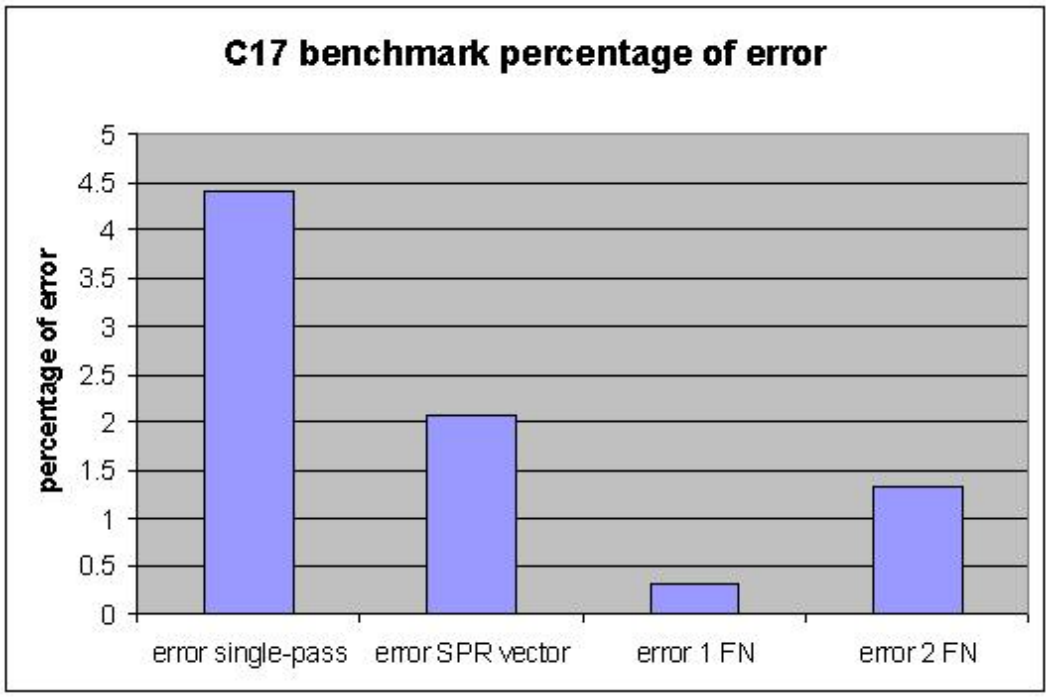

Figure 7.11: Errors for C17 benchmark

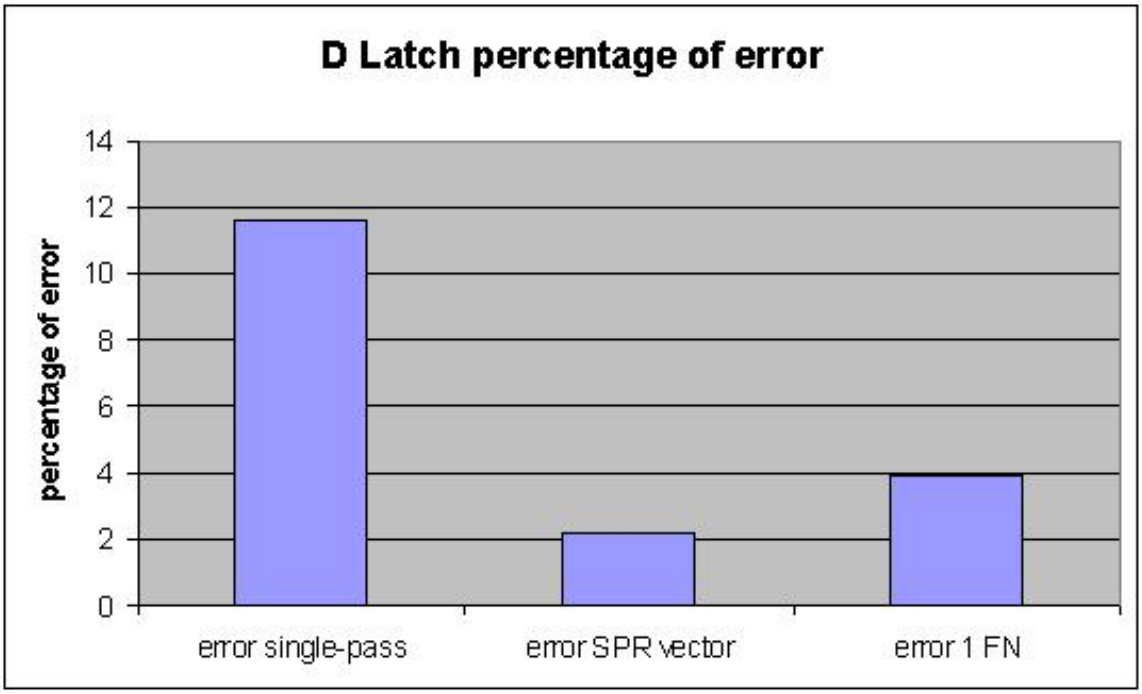

Figure 7.12: Errors for clocked D Latch

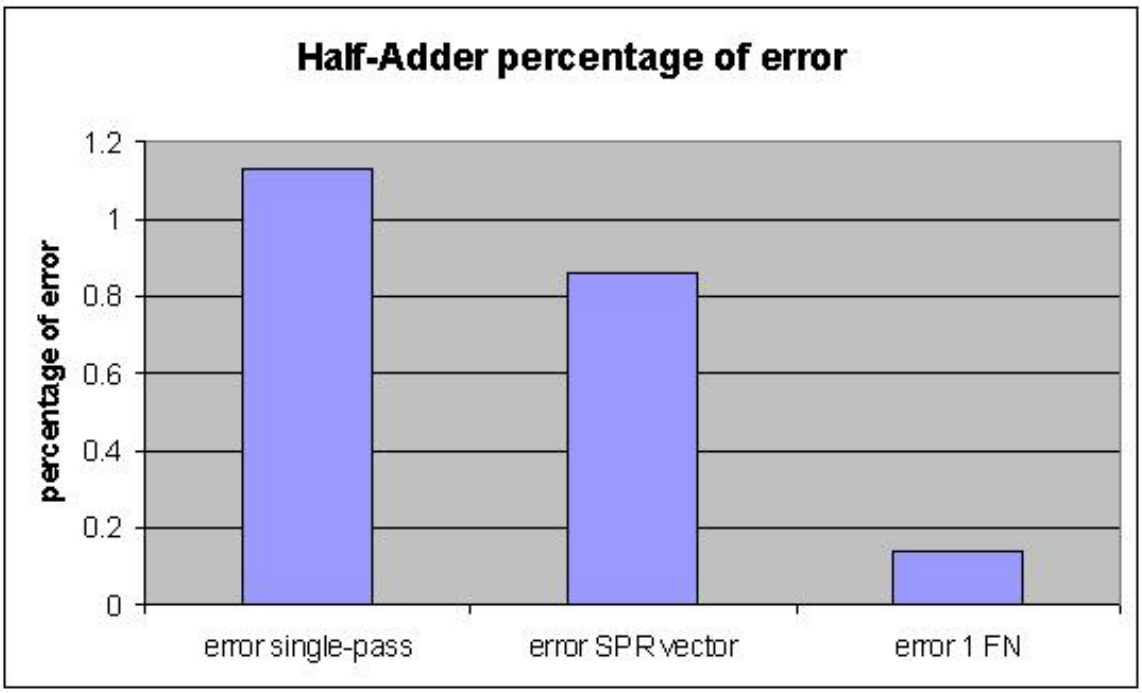

Figure 7.13: Errors for half-adder

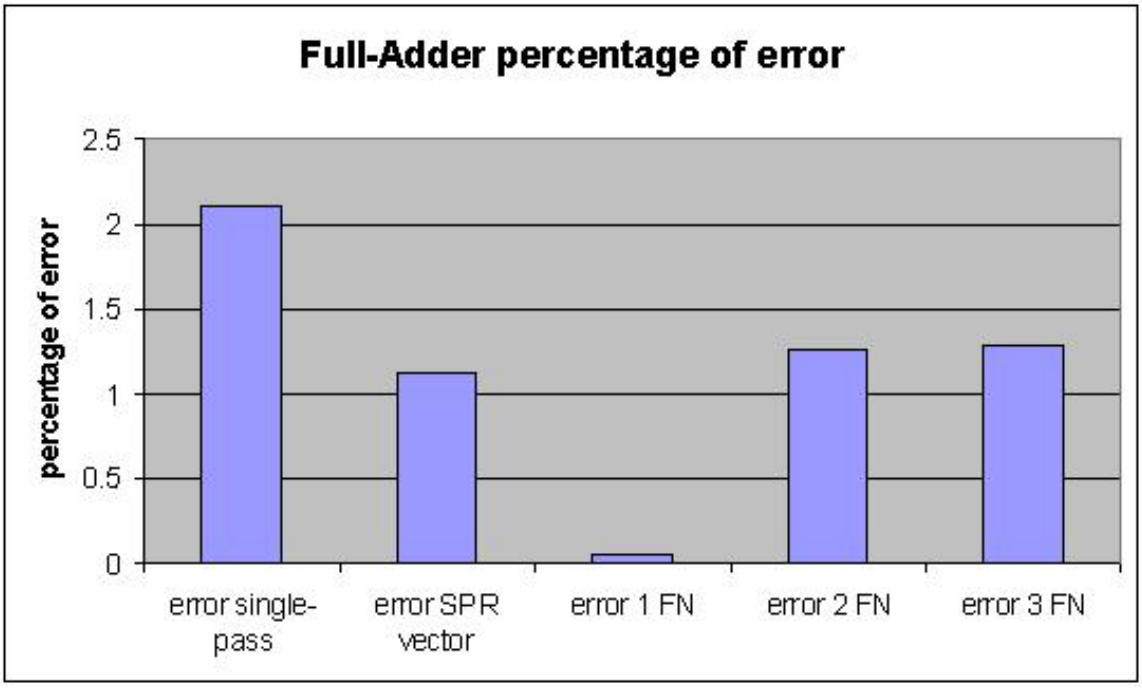

Figure 7.14: Errors for full-adder

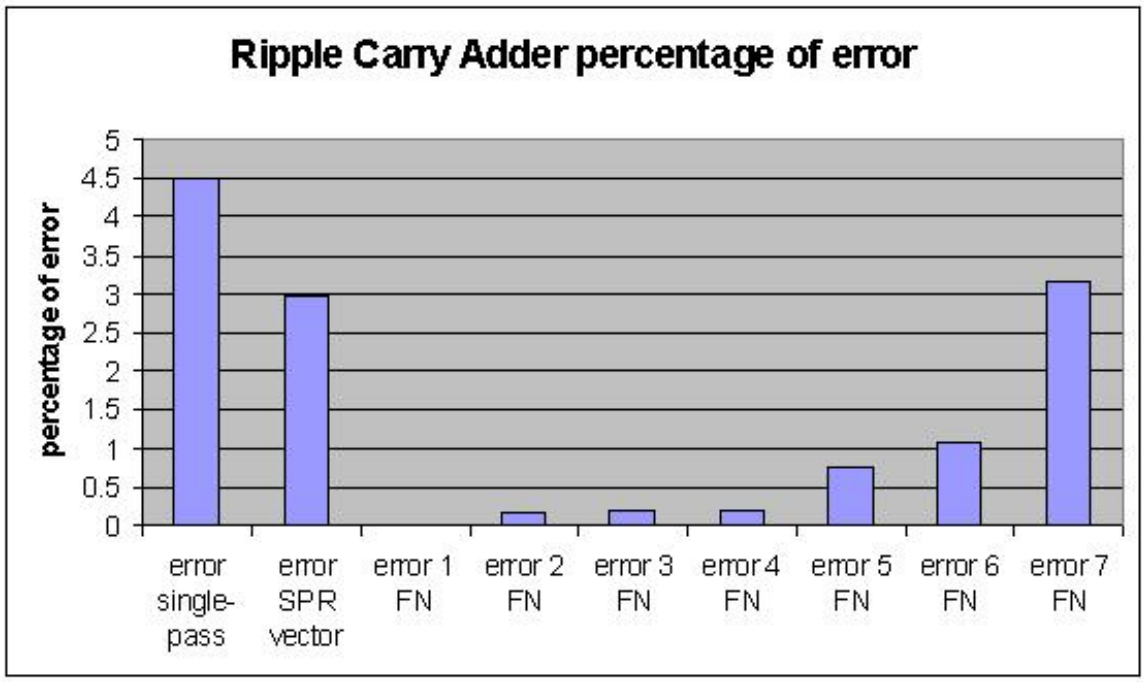

Figure 7.15: Errors for ripple-carry adder

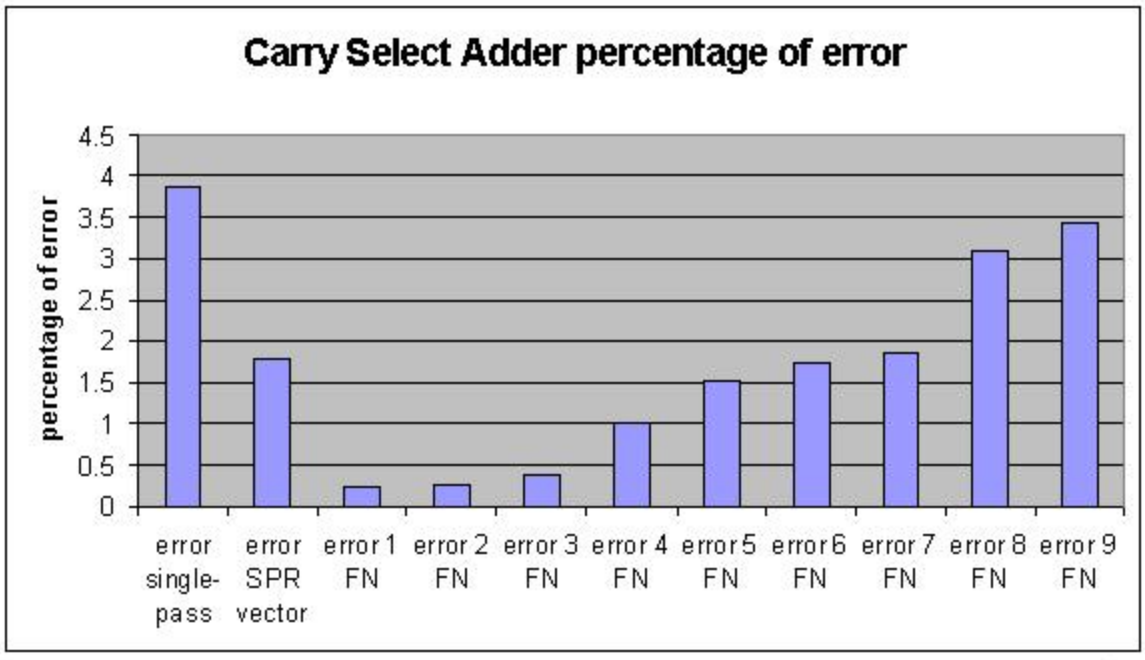

Figure 7.16: Errors for carry-select adder

3. depending on the topology of the circuit, its size and its number of fanouts and levels, the SPR vector optimization can be better than the SPR fanout neglection

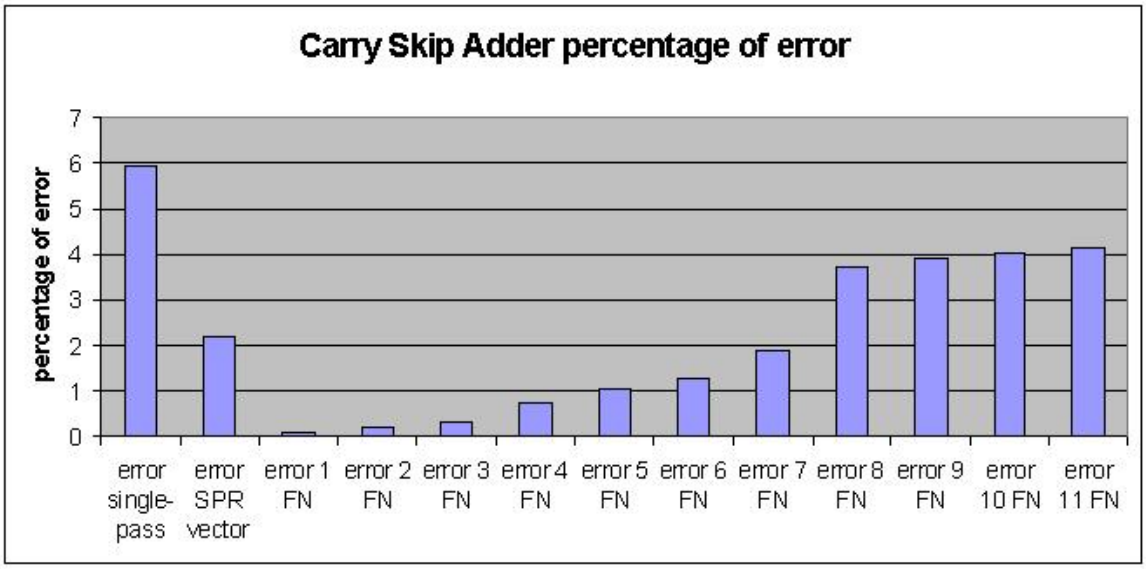

Figure 7.17: Errors for carry-skip adder

optimization, and vice versa.

7.4 Conclusion

At the end, we can say that the SPR vector and the SPR fanout neglection (SPR FN) can be used to compromise between the exact multi-pass algorithm that consume a lot in terms of memory usage and computing time, and the simple inexact single-pass algorithm. The optimization using the SPR fanout matrix simplification (SPR FMS) is rejected due to the fact that it gives less accurate results than the simple single-pass algorith of the SPR methodology for reliability analysis.

Chapter 8

Results: LMP model and tool for reliability analysis

8.1 Introduction

As seen in the introduction, integrated circuits will have a constant evolution with increases in density and speed. As CMOS technology is reaching its scaling limit, fault-tolerance of nanoscale devices is highly decreasing.

As a matter of fact, new nanotechnology CMOS devices will have a higher probability of being defective than current ones. The reliability of integrated circuits has thus become a key consideration when dealing with nanoscale designs.

As seen in the section concerning the state of the art, number of methodologies for evaluating circuit reliability were proposed in recent years.

Krishnaswamy and Patel presented the classical model to compute the reliability of logic circuits based on Probabilistic Transfer Matrix (PTM) [16, 5].

Han developed the Probabilistic Gate Model (PGM) of individual gates to iteratively build the reliability of its output [6].

Bhaduri and Norman introduced the Probabilistic Model Checking (PMC) that is based on Discrete Time Markov Chains (DTMC) [7, 8].

Correia presented the Probabilistic Binomial Reliability model for Reliability analysis (PBR) [10].

Franco developed another methodology to compute logic circuit reliability based on signal probabilities (SPR) [11].

Another method is the Weighted Averaging Algorithm (WAA) from the work of Krishnamurthy and Tollis [13].

Ercolani continued this work by proposing two heuristics for correcting signal correlation effects, the dynamic WAA (DWAA) and the correlation coefficients method [14].

Despite all these algorithms developed, a problem of scalability is commonly present for all methodologies. In fact, very huge calculations must be made just to obtain the reliability of large circuits.

To deal with this problem of scalability, we propose in this work a new methodology based on Level Matrix Propagation (LMP), consisting of iteratively propagating level matrix probabilities throughout all the circuit. Moreover, we present a new approach consisting of controlling reliability at each iteration.

8.2 PTM-based reliability analysis model

The reliability can be obtained using what is called a Probabilistic Transfer Matrix (PTM) specified for each gate of the logic circuit. PTM models the signal probability at the output according to the probability of inputs and the gate reliability value.

Figure 8.1 shows an example of the available PTM for an NAND2 logic gate, where q is the NAND2 logic gate reliability value and $1-q$ is the NAND2 logic gate probability of error.

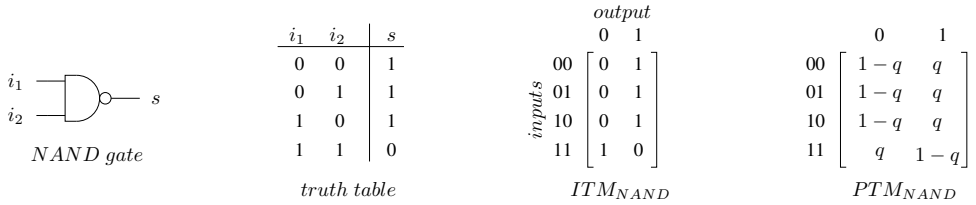

Figure 8.1: PTM and ITM representation of an NAND2 logic gate

A fault-free gate, i.e. a gate with a reliability equal 100%, is then represented by a PTM matrix with a gate reliability value 100%, which is called an Ideal Transfer Matrix (ITM). The reliability of the whole circuit is simply determined by the circuit PTM and the circuit ITM. Given a circuit e, with PTM T and ITM I, the reliability can be determined according to the expression in (8.1).

$$R(e) = \sum_{I(i,j)=1} p(i|j)p(i) \tag{8.1}$$

The $p(i)$ element represents the probability of an input value i, and $p(i|j)$ is the $(i,j)th$ element in T representing the probability correlation of output j given an input

i.

The PTM for systems with several gates is obtained as following:

1. PTM of gates in parallel is given by kronecker product of individual gates's PTM
2. PTM of gates in series is given by inner product of of individual gate's PTM.

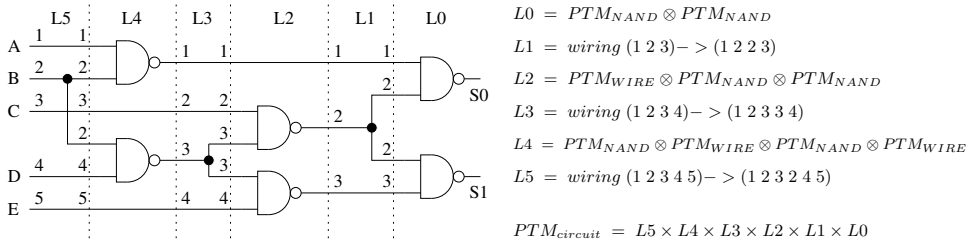

Figure 8.2: PTM of the C17 from ISCAS'85 benchmark

Figure 8.2 shows an example of PTM computation for the circuit C17 from ISCAS'85 benchmark [17]. The circuit is organized in six levels. For each level kronecker products are used to obtain the corresponding PTMs (L0 to L5), then inner products of such PTMs give the circuit PTM.

The main advantages of the PTM model are its straightforward implementation and its accuracy. The main disadvantage of this method is the scalability problem.

In fact, the PTM of a circuit is a matrix representation of input and output pattern possibilities, so the size of this matrix increases exponentially with the number of inputs and outputs, what leads to intractable computing times and memory storage needs for practical circuits.

8.3 LMP model

8.3.1 LMP methodology algorithm

Similarly to a scalable technique for reliability analysis developed by Bhaduri in [7], the algorithm of the LMP model would consists of:

1. topologically partitioning the full circuit by doing a levelized-list of all the circuits' gates
2. iteratively building the PTM and input matrix of each partition
3. propagation of the input matrix throughout the partition and use it as an input for the following partition

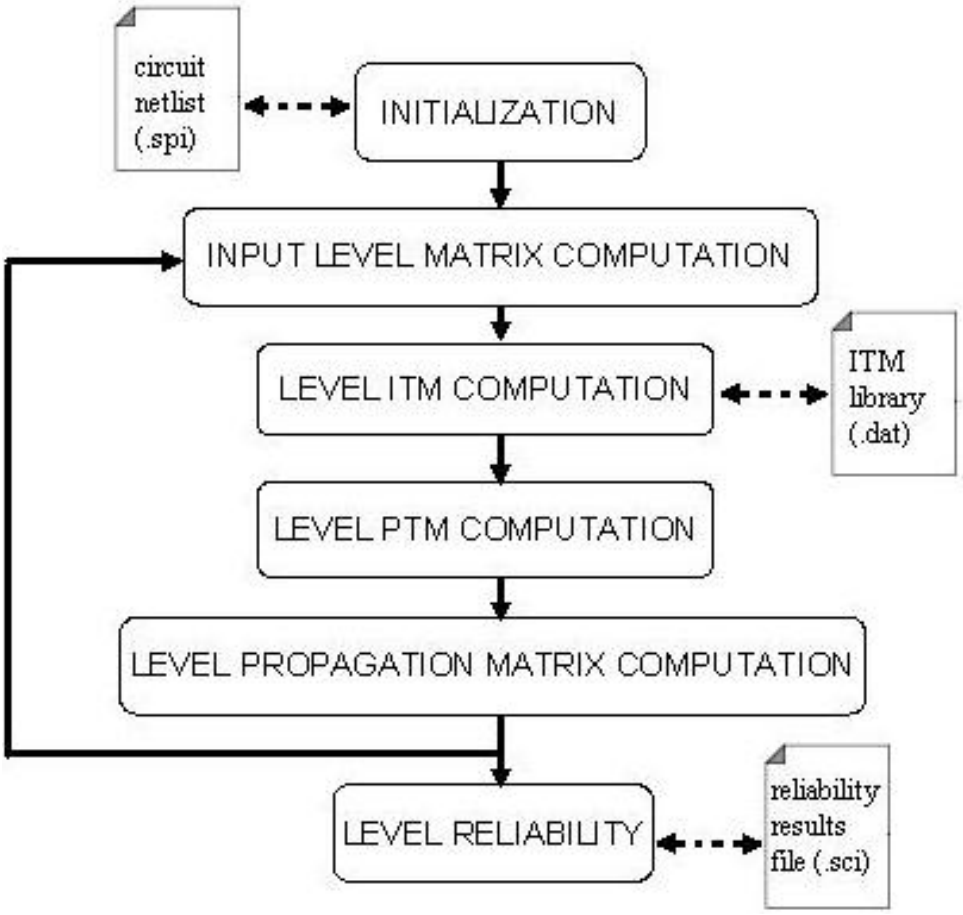

Figure 8.3: LMP scalable tool flow

4. clearing the preceding level matrices from the memory

Figure 8.3 shows the flow of the LMP scalable tool. A first phase of *Initialization* consists of transforming the circuit netlist (extracted from the VHDL/Verilog circuit description file) into a levelized matrix M of elements, where the columns represent the levels of the circuit and the rows represent the elements (gates or wires) of the circuit in top-down order.

An example of this levelized matrix for Figure 8.2 is shown in (8.2).

$$M = \begin{bmatrix} wire & nand & wire & wire & wire & nand \\ wire2 & wire & wire & nand & fanout & nand \\ wire & nand & fanout & nand & wire & 0 \\ wire & wire & wire & 0 & 0 & 0 \end{bmatrix} \tag{8.2}$$

Using this levelized matrix desciption of the circuit, we compute for each level *i*:

1. the input matrix

2. the ITM using an ITM library of elements
3. the PTM
4. the propagation matrix by multiplying the input matrix by the PTM matrix
5. the reliability by comparing the propagation matrix with the ITM
6. the level *i+1* input matrix before clearing matrices of i), ii), iii), vi) and v)

Same steps are done throughout the whole circuit till the last circuit level.

8.3.2 Controlled reliability calculation (CRC) approach

We propose a new approach of controlling reliability level by level for the LMP tool. By using this method, if the level reliability is going below the required circuit reliability, the designer can choose to stop the reliability computation and use another architecture for his circuit.

Figure 8.4 shows the example design of an adder with required circuit reliability equal to 96%.

Suppose the designer has the choice among 3 adder architectures, with increasing sizes: ripple carry, carry skip and carry select. The designer begins with a ripple carry adder because it requires less area, but the analysis is stopped at $t_{ripple_{L6}} = 377.09ms$ (using a 1.60GHz processor with 1.24GB of RAM) when the reliability crosses the 96% reliability required curve at *level 6*. The designer chooses next to use a carry skip adder but stops directly at *level 8* at $t_{skip_{L8}} = 199.23ms$ for the same reason. The designer then finally chooses a carry select adder and finds that the circuit reliability respects the requirements. Thus, using the LMP model, the total time required for design is:

$$t_{total_{LMP}} = t_{ripple_{L6}} + t_{skip_{L8}} + t_{select_{full}} = 1.14s$$

However, using a PTM model, the reliability analysis would have take the full reliability analysis time for the three adder architectures, which is:

$$t_{total_{PTM}} = t_{ripple_{full}} + t_{skip_{full}} + t_{select_{full}} = 1.46s$$

We can see that CRC can improve the reduction in run-time by stopping the reliability analysis process before ending the whole circuit. By the way, it can be efficient for the LMP tool to be integrated in the design flow of any logic circuit.

8.4 Comparison LMP vs. scalable PTM

To evaluate the functionality of some fault-tolerant approaches a library of arithmetic circuits, adders can be used. This library already exists and is used in this work.

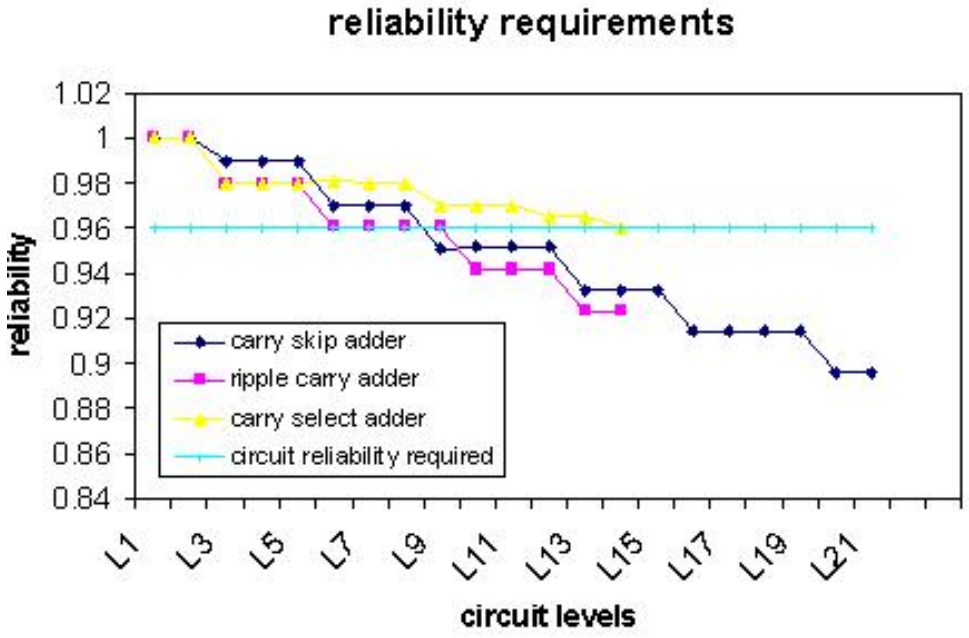

Figure 8.4: reliability requirements

These circuits carry well known tradeoffs between area, speed and power. Here, we make a further characterization of these circuits by means of signal reliability for a more detailed characterization of the respective design space.

The circuits in the library are described in the VHDL/Verilog languages, with parameterized data widths, and targeted to standard cell synthesis. This library can then extract spice netlists that can be used to compute circuit reliability using the scalable LMP model.

8.4.1 Memory usage: LMP vs. scalable PTM

At each iteration step, the complexity of the PTM matrix for the LMP model is smaller than the complexity of the PTM matrix for a PTM model.

In fact, for the LMP model, the PTM matrix complexity always equals the corresponding PTM level matrix complexity, while for the PTM model, the complexity grows exponentially from level to level. Compared to the monolithic scalable PTM approach, the technique performs better in terms of memory space.

Figure 8.5 shows the memory usage by the LMP tool versus the scalable PTM tool for four different topologies of adders from the library.

We easily notice that the LMP model improves the reliability calculation in terms of memory usage. For instance, for the carry skip adder, we obtain 6.45% reduction of

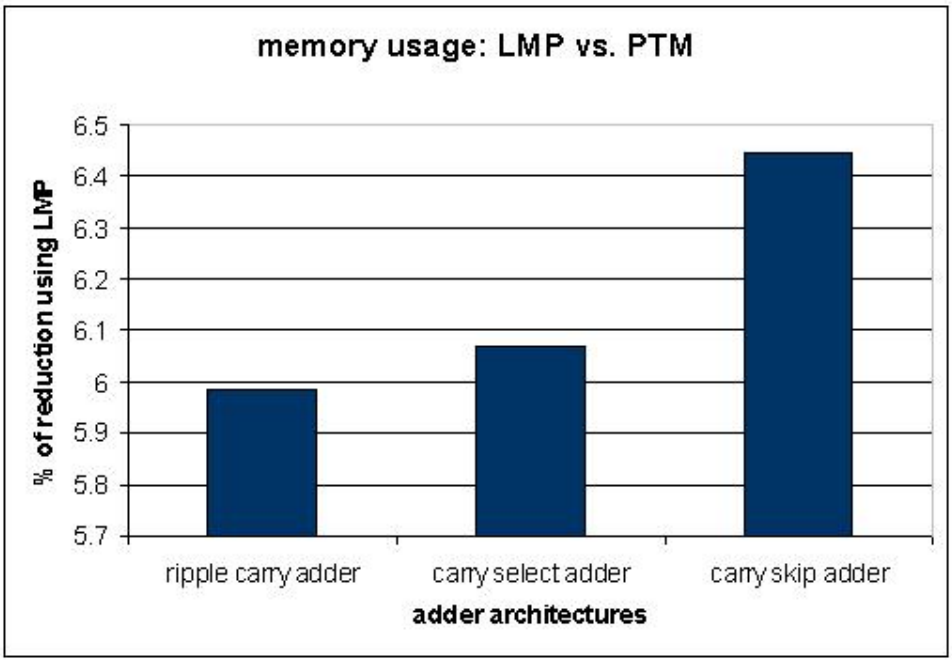

Figure 8.5: LMP vs. scalable PTM memory usage

the total memory used by the PTM model.

8.4.2 Run-time: LMP vs. scalable PTM

The LMP model propagates the level matrix, represented by the input matrix of the first level, from level to level througout all the circuit. But at each iteration step, in contrary to the PTM model, the LMP model propagates the input matrix from level *i* to level *i+1*.

Figure 8.6 shows the run-time by the LMP tool versus the scalable PTM tool for the same topologies of adder of the library.

While a bigger run-time is noticed for the full circuit with the LMP model, using the CRC approach, the designer makes the LMP model much more efficient in terms of computation run-time as can be seen in the previous section.

8.5 Conclusions

We addressed the problem of reliability analysis of logic circuits. We have developed a methodology and its tool, the LMP, that can be easily integrate as a loop in the design flow of logic circuits.

The LMP tool is scalable and uses a technique based on computing and propagating reliability values in levels, exploiting the structural decomposition of circuits, and controlling levels reliability. Thus, by calculating reliability of levels one by

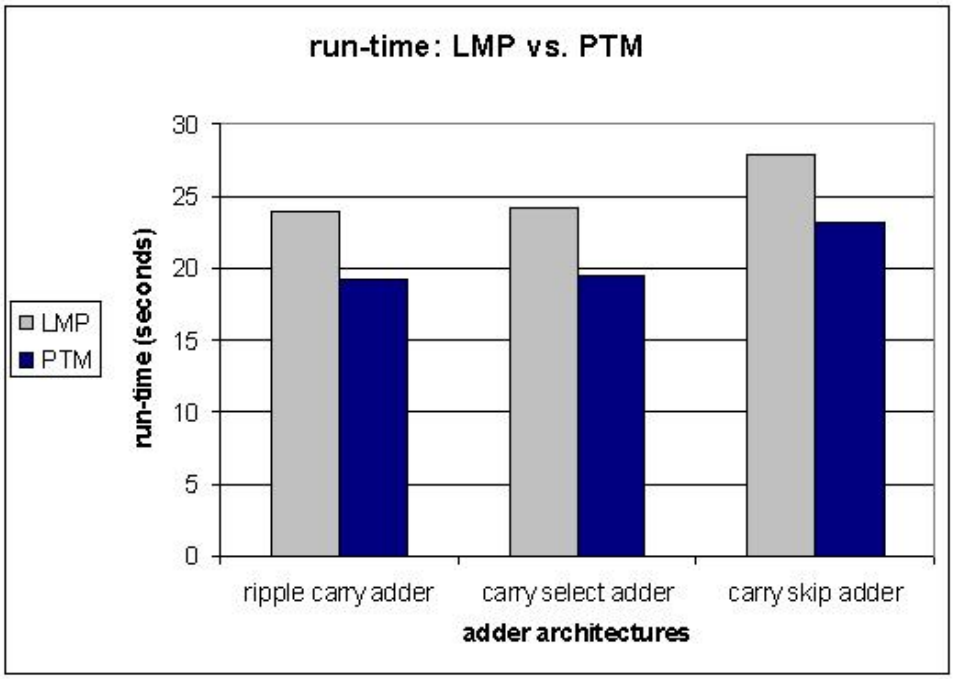

Figure 8.6: LMP vs. scalable PTM run-time

one, the designer can have the choice to stop the computation if the reliability isn't corresponding to circuit reliability requirement, without having to scan the entire circuit.

Chapter 9

Procedure Definition Solution

In the current work, we have:

1. validate our simplifications approaches of the SPR methodology:

   Vector SPR

   Fanout Neglections

2. validate our proposed scalable reliability analysis tool based on PTM, and we have named it "Level Matrix Propagation" (LMP) model for reliability analysis.

We have proved that we can compromize the SPR model without deviating a lot from the multi-pass exact value using the SPR vector and the SPR fanout neglection.

In fact, in the first part of our work we have proved that 3 compromises of the SPR methodology can be used to reduce the complexity of calculation. We have then validate the fact that we can use them to reduce the complexity of calculation without deviating a lot from the multi-pass algorithm. The limit of deviation was represented by the inexact circuit reliability value obtained using the single-pass algorithm of the SPR methodology.

The second part of our work showed that the Vector SPR compromise and SPR FN (fanout neglections) compromise give better result in terms of reliability accuracy than the single-pass SPR methodology.

Moreover, we have developed a tool that is better in terms of computing time and memory usage than the most classical reliability analysis model: the PTM model.

In fact, our tool called "Level Matrix Propagation" (LMP) is based on the propagation of the input matrices of each block from level to level, until scanning the entire circuit. LMP computes and display the reliability value of each level, and is controlled by the required reliability value given in input by the designer.

In the results part of our work, we have tested our LMP model on a set of adders used classically in the state of art of the reliability domain to test reliability analysis

models. The test have proved that our LMP model is better than the classical PTM model:

1. LMP model gives an exact circuit reliability value
2. LMP model is better than the PTM model in terms of memory usage
3. LMP model is worst than the PTM model in terms of run-time but our new approach of reliability control, the CRC, let us obtain better mean results than the PTM model in terms of run-time

Chapter 10

Conclusions

We have studied 3 possible optimizations of the SPR algorithm. We can conclude that the SPR Vector and the SPR fanout neglection can be used to compromise between the multi-pass algorithm and the single-pass algorithm.

We also addressed the problem of reliability analysis of logic circuits. We have developed a methodology and its tool, the LMP, that can be easily integrate as a loop in the design flow of logic circuits.

The LMP tool is scalable and uses a technique based on computing and propagating reliability values in levels. Thus, by calculating reliability of levels one by one, the designer can have the choice to stop the computation if the reliability isn't corresponding to circuit reliability requirement, without having to scan the entire circuit. The LMP is faster and consume less memory than the classical PTM model, the most used in the domain of reliability analysis.

Appendix A

SPR: Single-Pass Approach

This Appendix explicits the single-path approach of the signal probability-based reliability analysis.

Consider the circuit below of Figure B.1 composed of AND2 and invertor gates, supposed having a reliability of $q = 100\%$.

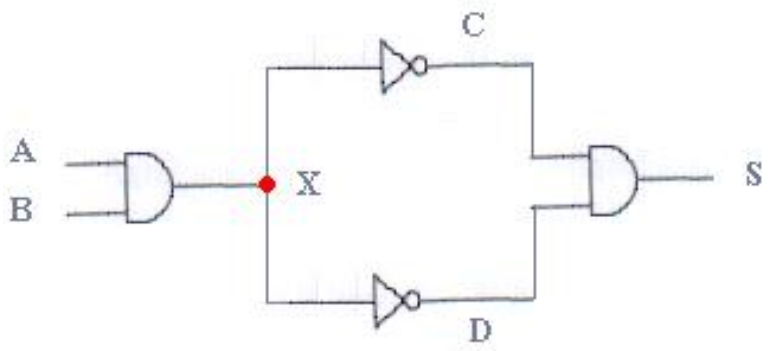

Figure A.1: Simple circuit having one reconvergent fanout

The single-pass methodology computes the circuit reliability R without taking into consideration the reconvergent fanout in the node X of the circuit. The circuit has 3 levels, let's compute signal probabilities of each level:

1. **Level 1:**

   Consider that inputs A and B have an equal probability of $correct0$ and $correct1$, then their signal probability matrices can be shown in B.2:

   $$A = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \; B = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \tag{A.1}$$

   The Input joint matrix I is the kronecker product of the inputs A&B and is computed as in B.3 below:

$$I = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \otimes \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} = \begin{bmatrix} 0.25 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0 \\ 0 & 0 & 0.25 & 0 \\ 0 & 0 & 0 & 0.25 \end{bmatrix} \tag{A.2}$$

We can have the Probabilistic Transfer Matrix (PTM) from the Ideal Transfer Matrix (ITM) of the AND2 gate as shown in B.4:

$$ITM = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \; PTM = \begin{bmatrix} q & 1-q \\ q & 1-q \\ q & 1-q \\ 1-q & q \end{bmatrix} \tag{A.3}$$

The propagation matrix P is the multiplication of the Input joint matrix I and the the PTM of the AND2 gate as shown in B.5. Having supposed all circuit gates reliability equal 100%, the PTM of the AND2 gate is simply the ITM of the AND2 gate.

$$P = I \times PTM = \begin{bmatrix} 0.25 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0 \\ 0 & 0 & 0.25 & 0 \\ 0 & 0 & 0 & 0.25 \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.25 & 0 \\ 0.25 & 0 \\ 0.25 & 0 \\ 0 & 0.25 \end{bmatrix} \tag{A.4}$$

Using the equations in B.6, A.6, A.7 and A.8:

$$p_0(signal) = \sum_{r:ITM(0,r)=1} P(signal)_{[0,r]} \tag{A.5}$$

$$p_1(signal) = \sum_{r:ITM(1,r)=0} P(signal)_{[1,r]} \tag{A.6}$$

$$p_2(signal) = \sum_{r:ITM(0,r)=0} P(signal)_{[0,r]} \tag{A.7}$$

$$p_3(signal) = \sum_{r:ITM(1,r)=1} P(signal)_{[1,r]} \tag{A.8}$$

we obtain the 2×2 output signal probability matrix of node X as shown below in A.9

$$X = \begin{bmatrix} 0.75 & 0 \\ 0 & 0.25 \end{bmatrix} \tag{A.9}$$

2. **Level 2:**

As the circuit is symetric the nodes C and D will have the same output signal probabilities. Let's calculate then output signal probability matrix of node C:

The input signal probability matrix is simply the output signal probability matrix of Level 1 as seen in A.10

$$I = X = \begin{bmatrix} 0.75 & 0 \\ 0 & 0.25 \end{bmatrix} \tag{A.10}$$

The PTM of the invertor gate is as in A.11

$$PTM = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{A.11}$$

The propagation matrix is then as in A.12

$$P = I \times PTM = \begin{bmatrix} 0.75 & 0 \\ 0 & 0.25 \end{bmatrix} \times \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0.75 \\ 0.25 & 0 \end{bmatrix} \tag{A.12}$$

Using the equations in B.6, A.6, A.7 and A.8, we obtain the 2×2 output signal probability matrix of node C and the symetric node D as shown below in A.13

$$C = \begin{bmatrix} 0.25 & 0 \\ 0 & 0.75 \end{bmatrix} \; D = \begin{bmatrix} 0.25 & 0 \\ 0 & 0.75 \end{bmatrix} \tag{A.13}$$

3. **Level 3:**

The level 3 is composed again of a AND2 gate that takes as inputs signals C and D. The Input joint matrix is their kronecker product as shown in A.14

$$I = \begin{bmatrix} 0.25 & 0 \\ 0 & 0.75 \end{bmatrix} \otimes \begin{bmatrix} 0.25 & 0 \\ 0 & 0.75 \end{bmatrix} = \begin{bmatrix} 0.0625 & 0 & 0 & 0 \\ 0 & 0.1875 & 0 & 0 \\ 0 & 0 & 0.1875 & 0 \\ 0 & 0 & 0 & 0.5625 \end{bmatrix} \tag{A.14}$$

Using the PTM of the AND2 gate, we compute the propagation matrix P as shown in A.15

$$P = I \times PTM = \begin{bmatrix} 0.0625 & 0 & 0 & 0 \\ 0 & 0.1875 & 0 & 0 \\ 0 & 0 & 0.1875 & 0 \\ 0 & 0 & 0 & 0.5625 \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.0625 & 0 \\ 0.1875 & 0 \\ 0.1875 & 0 \\ 0 & 0.5625 \end{bmatrix} \tag{A.15}$$

Using the equations in B.6, A.6, A.7 and A.8, we finally obtain the 2×2 output signal probability matrix of node S, the output of the whole circuit as shown below in A.16

$$S = \begin{bmatrix} 0.4375 & 0 \\ 0 & 0.5625 \end{bmatrix} \tag{A.16}$$

The circuit Reliability R is the sum of the $correct0$ and the $correct1$ as in A.17

$$R = 0.4375 + 0.5625 = 1 \tag{A.17}$$

As predicted, the circuit reliability is 100% as all the circuit reliability gates are 100%.

Figure A.2 the circuit reliability values R with regards to the gates reliability values q

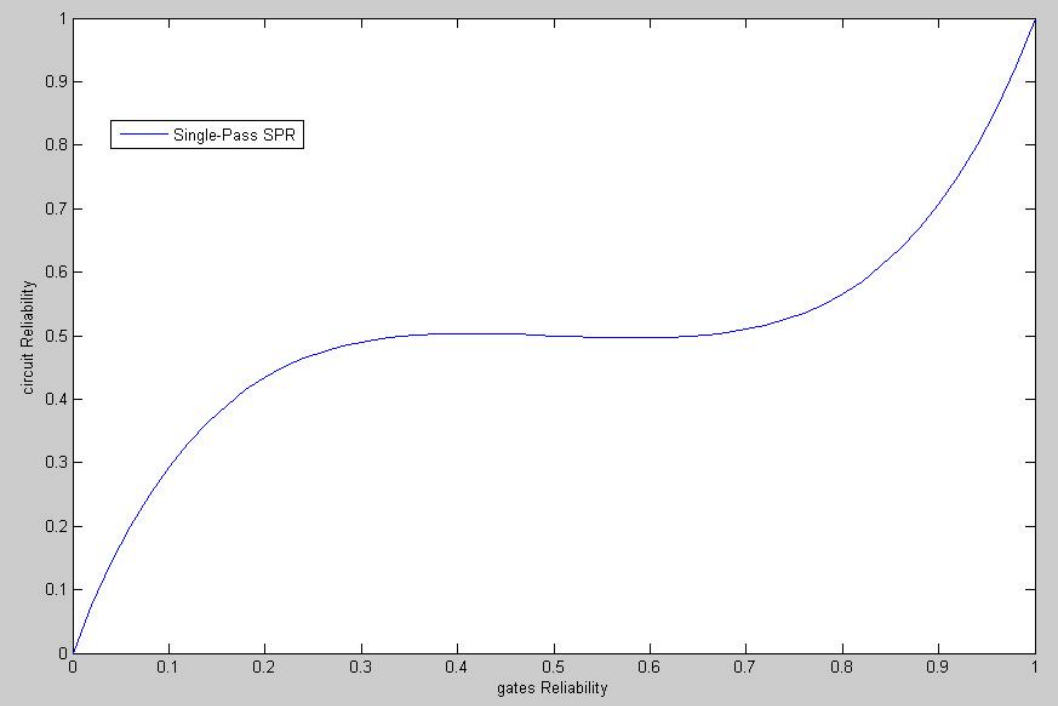

Figure A.2: gates reliability vs circuit reliability for Single-Pass SPR

Appendix B

SPR: Multi-Pass Approach

Appendix A presented the Single-Pass approach of the SPR model for reliability analysis that does not take into account the reconvergent fanout of the circuit. Let's illustrate the Multi-Pass approach of the SPR which gives the exact circuit reliability value as it takes into account reconvergent fanouts of the circuit, and compare it with the Single-Pass.

Consider again the circuit below of Figure B.1 composed of AND2 and invertor gates, supposed having a reliability of $q = 100\%$.

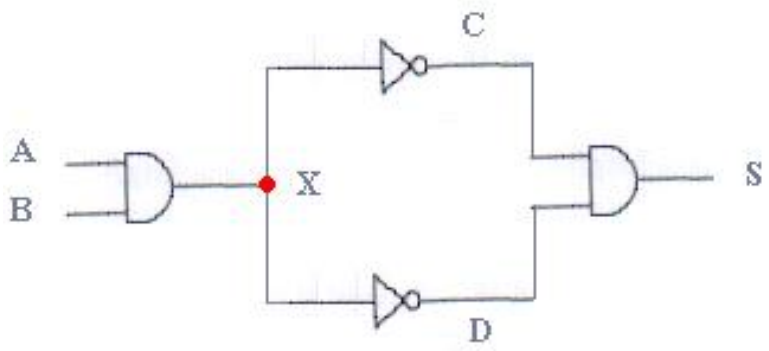

Figure B.1: Simple circuit having one reconvergent fanout

The multi-pass methodology computes the circuit reliability R taking into consideration the reconvergent fanout in the node X of the circuit. As the circuit has one reconvergent fanout, the number of iterations are $4^1 = 4$.

The single-pass methodology is normally used until the fanout signal of the circuit. We can then use the signal X matrix obtain in the Appendix A of the single-pass approach in B.1

$$X = \begin{bmatrix} 0.75 & 0 \\ 0 & 0.25 \end{bmatrix} \tag{B.1}$$

1. **Iteration 1:**

   The iteration 1 is simply a single-pass circuit reliability computation considering only the first signal probability of the fannout signal X.

   Figure B.2 shows the considered signal X matrix during this first iteration.

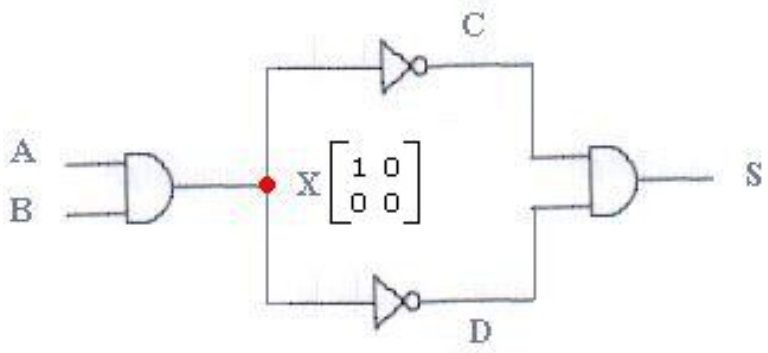

Figure B.2: Iteration 1 of the Multi-Pass SPR

Using the single-pass methodology we obtain a 2×2 output signal probability matrix of the circuit o_1 as shown below in B.2

$$o_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{B.2}$$

This matrix is then ponderated by the signal probability of a $P(correct0)_X = 0.75$ of the node X, and we obtain the matrix S_1 of the first iteration.

The reliability $R_1 = 0.75$

2. **Iteration 2:**

   Figure B.3 shows the considered signal X matrix during this first iteration.

   Using the single-pass methodology we obtain a 2×2 output signal probability matrix of the circuit o_2 as shown below in B.3

$$o_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{B.3}$$

This matrix is then ponderated by the signal probability of a $P(incorrect1)_X = 0$ of the node X, and we obtain the matrix S_2 of the first iteration.

The reliability $R_2 = 0$

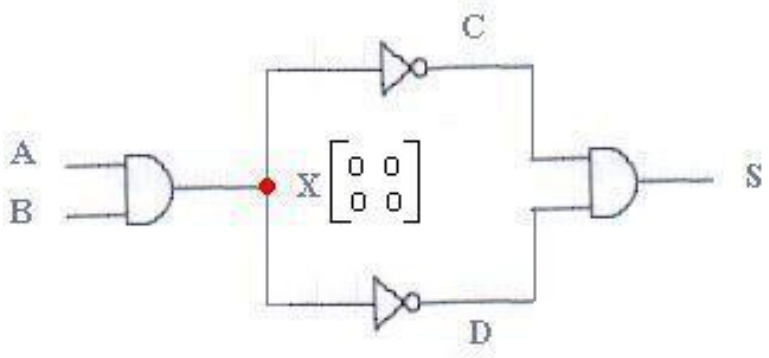

Figure B.3: Iteration 2 of the Multi-Pass SPR

3. **Iteration 3:**

   Figure B.4 shows the considered signal X matrix during this first iteration.

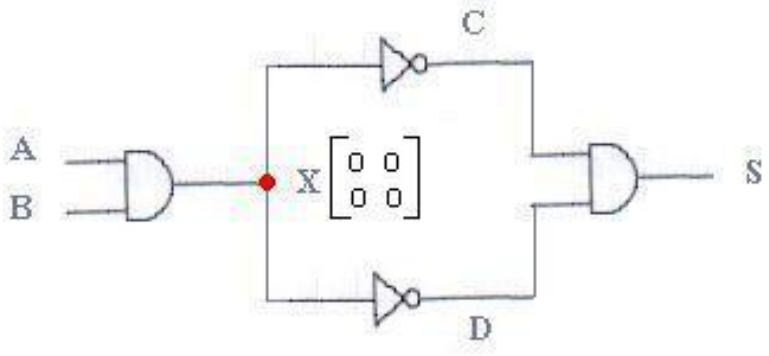

Figure B.4: Iteration 3 of the Multi-Pass SPR

   Using the single-pass methodology we obtain a 2×2 output signal probability matrix of the circuit o_3 as shown below in B.4

$$o_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{B.4}$$

   This matrix is then ponderated by the signal probability of a $P(incorrect0)_X = 0$ of the node X, and we obtain the matrix S_3 of the first iteration.

   The reliability $R_3 = 0$

4. **Iteration 4:**

   Figure B.5 shows the considered signal X matrix during this first iteration.

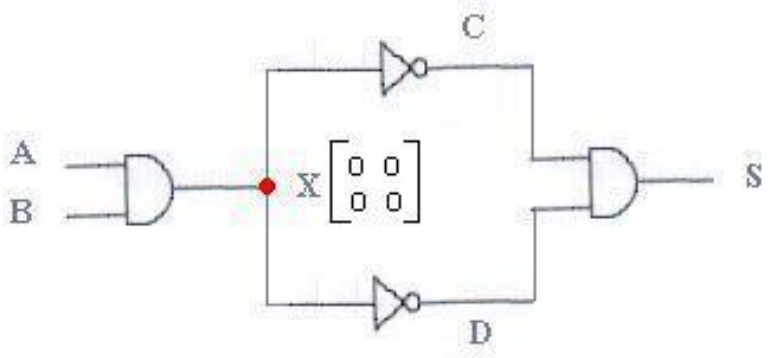

Figure B.5: Iteration 4 of the Multi-Pass SPR

Using the single-pass methodology we obtain a 2×2 output signal probability matrix of the circuit o_4 as shown below in B.5

$$o_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \tag{B.5}$$

This matrix is then ponderated by the signal probability of a $P(correct1)_X = 0.25$ of the node X, and we obtain the matrix S_4 of the first iteration.

The reliability $R_4 = 0.25$

5. **Circuit reliability computation:**

To obtain the final circuit reliability computation, we only have to sum the circuit reliability of each iteration as shown in B.6

$$R = R_1 + R_2 + R_3 + R_4 = 1 \tag{B.6}$$

As predicted, the circuit reliability is 100% as all the circuit reliability gates are 100%.

Figure B.6 the circuit reliability values R with regards to the gates reliability values q

6. **Comparison Multi-Pass vs Single-Pass:**

Figure B.7 represent the curves of the reliabilities of Multi-Pass and the Single-Pass versus the gate reliabilities.

We can also compare the Multi-Pass results with regards to the Single-Pass results by representing the area under the curve of the error e. This error is equal to the substraction of the area under the curve of the exact multi-pass reliabilities

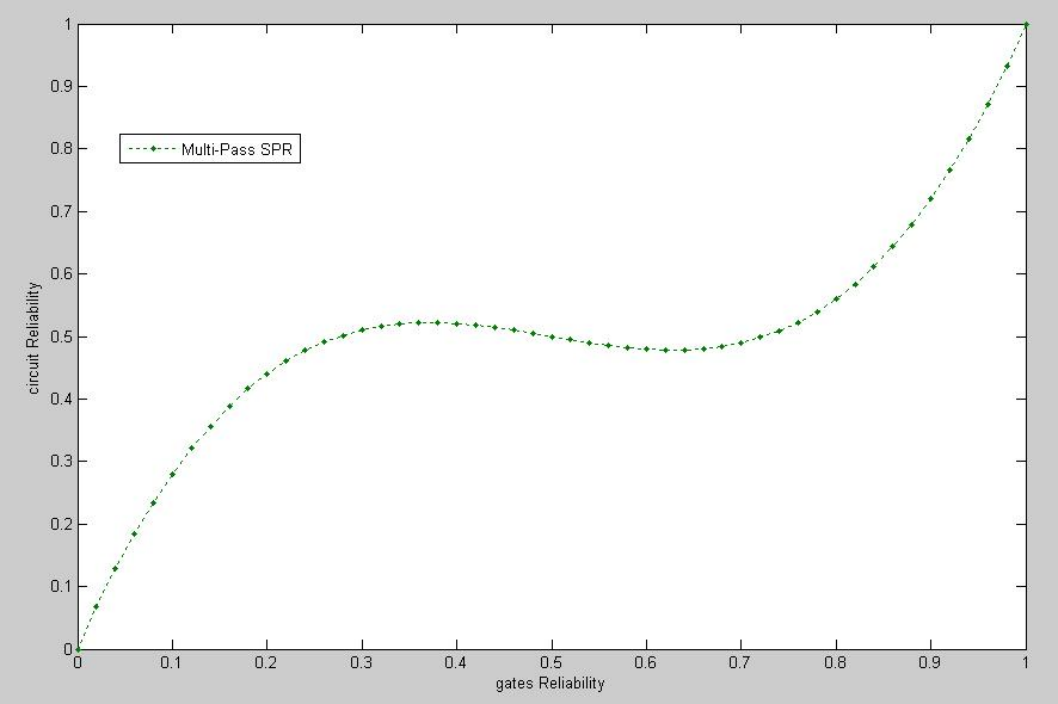

Figure B.6: gates reliability vs circuit reliability for Multi-Pass SPR

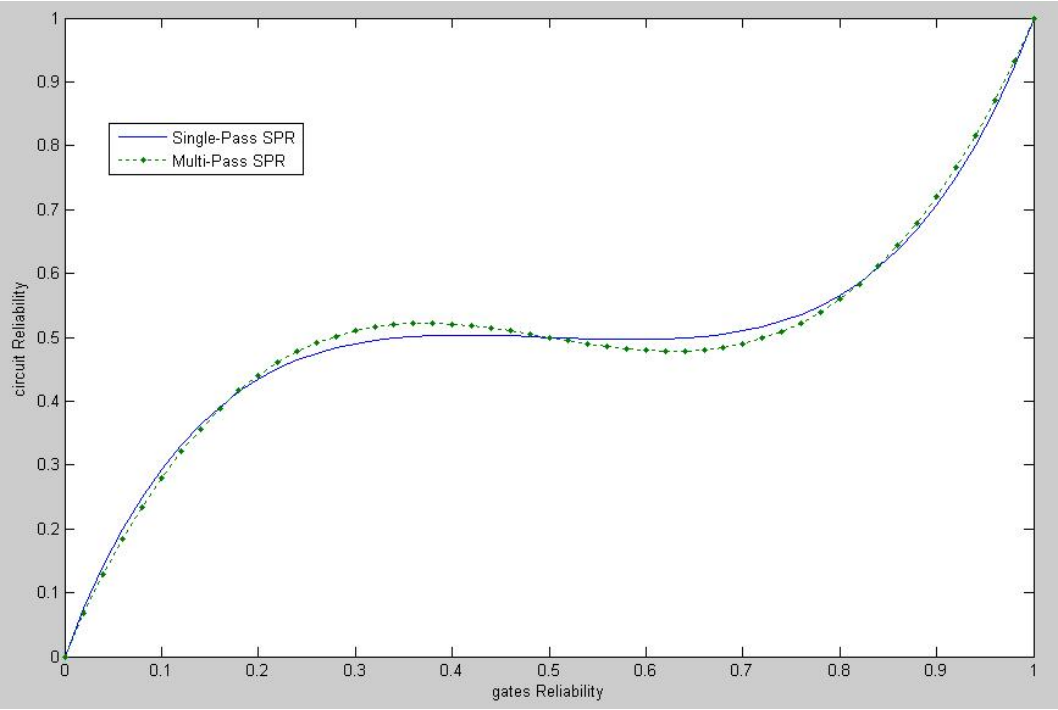

Figure B.7: gates reliability vs circuit reliability for Multi-Pass SPR and Single-Pass SPR

from the non-exact single pass reliabilities, and we obtain the curve of Figure B.8.

The curve illustrate the gates reliability values where single-pass and multi-pass gives the same circuit reliability results. Moreover, we can see the gates reliability values where the error is maximum $e_{max} = 0.02$ or minimum $e_{min} = 0.015$.

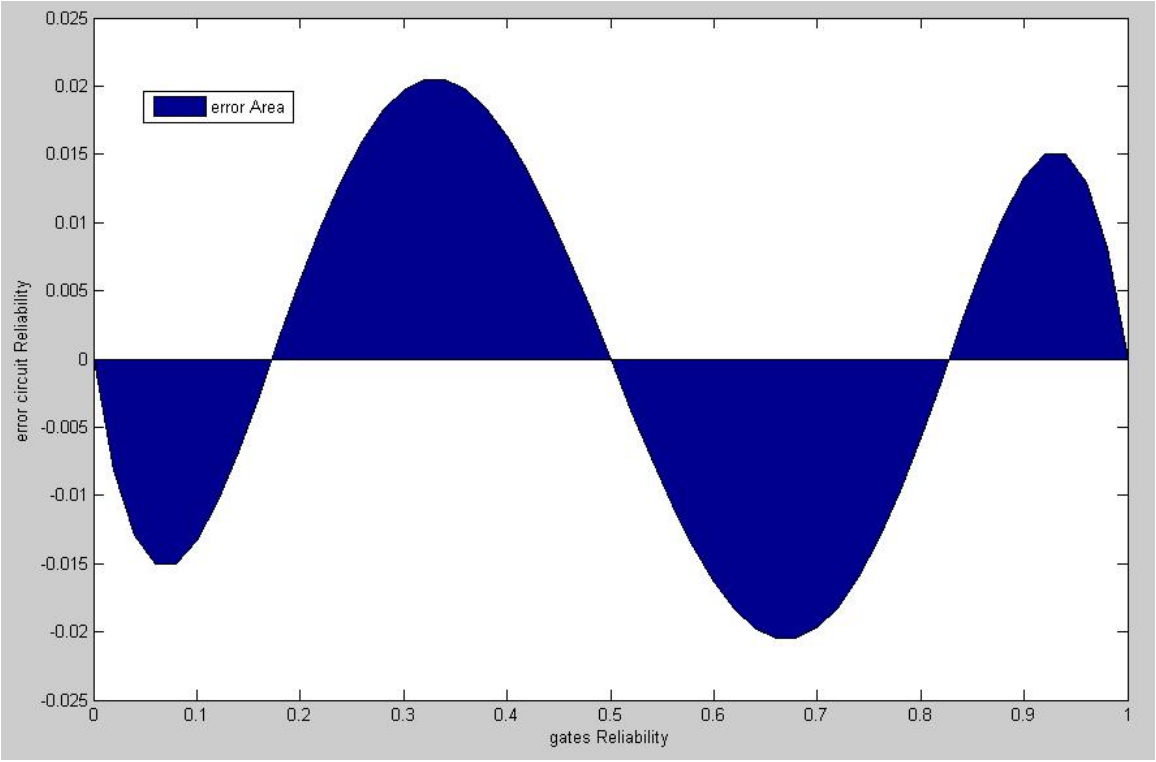

Figure B.8: error reliability Multi-Pass vs reliability Single-Pass

Bibliography

[1] E. Mollick. Establishing Moore's Law. *IEEE Annals of the History of Computing*, pages 62–75, July-September 2006.

[2] Denis Teixeira Franco, Jean-Francois Naviner, and Lirida Naviner. Yield and Reliability issues in Nanotechnologies. *Annales des telecommunications*, pages 3–4, November-December 2006.

[3] Way Kuo. Challenges Related to Reliability in Nano Electronics. *IEEE Transactions on Reliability*, 55(4):1–2, december 2006.

[4] Smita Krishnaswamy, George F. Viamontes, Igor L. Markov, and John P. Hayes. Accurate reliability evaluation and enhancement via probabilistic transfer matrices. *Proceedings of Design Automation and Test in Europe*, (1):282–287, 2005.

[5] Ketan N. Patel, Igor L. Markov, and John P. Hayes. Evaluating circuit reliability under probabilistic gate-level fault models. *Proceeding of the Twelfth International Workshop on Logic and Synthesis (IWLS 2003)*, 1:59–64, May 2003.

[6] Jie Han, E.Taylor, Jianbo Gao, and J.Fortes. Faults, error bounds and reliability of nanoelectronic circuits. *16th IEEE International Conference*, pages 247–253, July 2005.

[7] Debayan Bhaduri, Sandeep Shukla, Paul Graham, and Maya Gokhale. Scalable techniques and tools for reliability analysis of large circuits. *Proceedings of the 20th International Conference on VLSI Design, 2007*, pages 705–710, Jan. 2007.

[8] G. Norman, D. Parker, M. Kwiatkowska, and S. Shukla. Evaluating the reliability of NAND multiplexing with PRISM. *IEEE*, 24(10):1629–1637, October 2005.

[9] D.P. Siewiorek and R.S. Swarz. Reliable Computer Systems: Design and Evaluation. *Digital Press*, pages 263–278, 1992.

[10] Mai Correia, Denis Teixeira Franco, Lirida Naviner, and Jean-François Naviner. Reliability Analysis of Combinational Circuits Based on a Probabilistic Binomial Model. *NEWCAS-TAISA'08*, pages 310–313, June 2008.

[11] Denis Teixeira Franco, Mai Correia, Lirida Naviner, and Jean-François Naviner. Reliability of Logic Circuits Under Multiple Simultaneous Faults. *IEEE International MWSCAS 2008*, pages 265–268, Aug. 2008.

[12] Denis Teixeira Franco, Mai Correia, Lirida Naviner, and Jean-François Naviner. Signal Probability for Reliability Evaluation of Logic Circuits. *ESREF 2008*, pages 1587–1591, Sep.-Oct. 2008.

[13] B. Krishnamurthy and I.G. Tollis. Improved techniques for estimating signal probabilities. *Computers, IEEE Transactions on*, 38(7):1041–1045, Jul 1989.

[14] S. Ercolani, M. Favalli, M. Damiani, P. Olivo, and B. Ricco. Estimate of signal probability in combinational logic networks. *European Test Conference, 1989., Proceedings of the 1st*, pages 132–138, 12-14 Apr 1989.

[15] Mihir R. Choudhury and Kartik Mohanram. Accurate and scalable reliability analysis of logic circuits. *Proceedings of Design Automation and Test in Europe (DATE)*, pages 1454–1459, March 2007.

[16] Smita Krishnaswamy, George F. Viamontes, Igor L. Markov, and John P. Hayes. Accurate reliability evaluation and enhancement via probabilistic transfer matrices. *Proceedings of Design Automation and Test in Europe (DATE)*, 1:282–287, March 2005.

[17] Mark C. Hansen, Hakan Yalcin, and John P. Hayes. Unveiling the ISCAS-85 Benchmarks: A Case Study in Reverse Engineering. *IEEE Design and Test of Computers*, pages 72–80, jul-sept 1999.