Reliability Analysis of Logic Circuits Using Probablistic Techniques

Satish Grandhi, Christian Spagnol, Emanuel Popovici
Department of Electrical and Electronic Engineering,
University College Cork
Cork, Ireland
sagrand@ue.ucc.ie, christian.spagnol@ue.ucc.ie, e.popovici@ucc.ie

***Abstract*—The low reliability of advanced CMOS devices has become a critical issue that can potentially supersede the benefits of the technology shrinking process. This is making the design time reliability assessment and optimization a mandatory step in the IC design flow. As part of our ongoing research, we describe an algorithm based on probability analysis and logic principles for computing the impact of gate failures on the circuit output. We also propose a Bound and Propagate based methodology to handle the reconvergent fanout issue. A reliability evaluator has been developed around the open source logic synthesis tool 'abc' to allow for the integration and evaluation of our method in the context of an IC design flow. This approach had tremendously reduced the computation time while maintaining adequate precision. Simulation results for several benchmark circuits demonstrate the accuracy and the simulation time advantages when compared to MonteCarlo simulations.**

Keywords:Reliability, Reconvergent Fanout, AIG, abc

I. Introduction

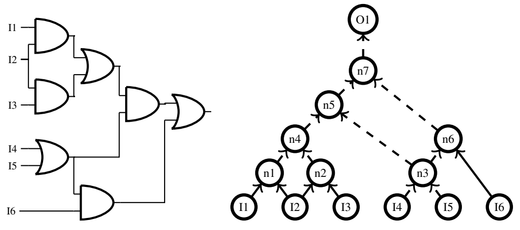
As CMOS technology enters the nanometer era they force the design tolerance limits to its lowest possible levels. Nanotechnology specific issues, e.g., V$_{dd}$ reduction, higher impact of process variation and temperature related aging resulted in increased device failure rates, making CMOS ICs less reliable [1], [2]. Therefore, design time reliability assessment and optimization are becoming a mandatory step in the IC design flow. As part of our ongoing research, we are developing a reliability aware logic synthesis tool. The first step in this process is to develop an efficient algorithm that computes circuit reliability.

Reliability analysis of logic circuits deals with computing the impact that the errors gate levels has on the circuit Primary Outputs. Traditional approach to reliability analysis begins with elementary SPICE simulations to estimate the circuit error probability. Several analytical approaches for computing reliability have been previously reported [3], [4]. As we represent circuits in the AIG format, a novel algorithm based on probability principles is proposed, with the prime focus being AND & Inverter gates. The algorithm uses the dynamic weighted average algorithm (DWAA) [5] approach to estimate the impact of reconvergent fanout on Static Probabilities. Further, Bound and Propagate methodology is introduced to account for the statistical dependence on error probabilities due to reconvergent fanout.

This paper is organised as follows. Section II, presents the data structure used in this work. Section III discusses the mathematics behind the error model. Section IV demonstrates the implementation, along with the design flow incorporating the tool. Section V introduces the error bounds. Section VI concludes and outlines our ongoing research effort.

II. And Inverter Graph

Structural representation, logic synthesis and technology mapping of a Boolean function are important issues in the design of digital circuits. One of the major decision in designing an EDA tool is selection of the right data structure as it determines the speed and efficiency of the tool. Binary Decision Diagrams (BDDs) has been used extensively but they have reached the limit of their scalability due to the always increasing complexity of modern circuitry.



(a) Reference Circuit *(b)* AIG representation

Fig. 1. And Inverter representation of Combinational Circuit

And Inverter Graph (AIG) [6] is a Boolean network composed of two-input-ANDs and inverters. Fig. 1 depicts a simple combinational circuit and its corresponding AIG representation. The circle represent the 2-Input AND gates and edges with a dash line indicate negation i.e. inversion of that input. AIG unifies equivalence checking, synthesis and technology mapping and offer better performance and correlation with final area and delay once the circuit has been mapped to a target technology [7], [8], [9]. We believe that the AIG data structure is suitable for representing combinational circuits also from a reliability prospective. In particular the fact that AIG are non canonical (i.e.: there exist more graphs representing the same logic function) can be exploited to improve reliability Moreover such a property may allow easier/faster implementation of error coding graph augmentation techniques. The first step is to develop probabilistic gate error models for AND & inverter logic gates.

Some of the common conventions used through out the paper are listed below:

- P_1(Z) - Probability of node 'Z' to be 1
- P_0(Z) - Probability of node 'Z' to be 0
- P_ϵ(Z) - Probability of error on node 'Z'
- P_f(y) - Failure probability of Logic Gate 'y'
- R(y) - Reliability of Logic Gate 'y'

III. Gate Error Models

An unreliable AND gate can be modeled as an ideal (error free) AND gate followed by a faulty buffer that represents the stochastic behavior of the errors. This model moves the entire error statistic on the output and so it implicitly assumes a symmetrical error behavior in relation to the inputs. The two nodes Z* and Z are named as internal and the external output node. Next, the analysis of the output error due to two possible reasons: (i) propagation of the errors onto the gate input nodes and (ii) intrinsic errors within the gate, is presented.

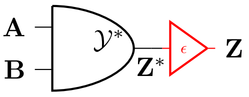


Fig. 2. Unreliable AND Gate Model

A. 2-Input Ideal AND Gate

As AIG's comprises of only 2-input AND gates, we restrict our analysis to 2 input AND gates and note that its extension to multi-input AND gates is straightforward. The static probability values of the internal output node Z* can be expressed as:

$$P_{z^*}(1) = P[A = 1, B = 1] \tag{1}$$

We note that error on the inputs need not necessarily result in a wrong output value. Consider error free '0' on one of the input pin. This would mask the error on the other input node from being propagated onto the output. Similarly, consider the scenario where the input pins are set to '0' & '1' and both are in error. This event of double error mutually negates each other and will still result in a correct output state. Hence, due to the masking and double error events, there are no simple rules to predict the state of the output. Tab. I presents an exhaustive enumeration on all the possible cases with the associated output status. To explain the table, consider the case of input A=0 and B=0. Then, the internal output Z* evaluates to '0'. Now, each of the inputs can be in error (ϵ) or correct (c) state. The state of the inputs determines if Z*=0 is correct or not. It is clear that, for these inputs values, the internal output node is in error if and only if both the inputs are in error. It is evident from Tab. I that only 6 of the possible 16 cases result in error on the internal output. The probabilities for each of these events to occur are presented in the last column.

Reliable Condtions			Unreliable Condtions			Error Probability
A	B	Z*	A	B	Z*	
0	0	0	c/0	c/0	c/0	
			c/0	ϵ/1	c/0	
			ϵ/1	c/0	c/0	
			ϵ/1	**ϵ/1**	**ϵ/1**	$\mathbf{P[A=0,B=0,A_\epsilon,B_\epsilon]}$
0	1	0	c/0	ϵ/0	c/0	
			c/0	c/1	c/0	
			ϵ/1	ϵ/0	c/0	
			ϵ/1	**c/1**	**ϵ/1**	$\mathbf{P[A=0,B=1,A_\epsilon,B_c]}$
1	0	0	ϵ/0	c/0	c/0	
			ϵ/0	ϵ/1	c/0	
			c/1	c/0	c/0	
			c/1	**ϵ/1**	**ϵ/1**	$\mathbf{P[A=1,B=0,A_c,B_\epsilon]}$
1	1	1	**ϵ/0**	**ϵ/0**	**ϵ/0**	$\mathbf{P[A=1,B=1,A_\epsilon,B_\epsilon]}$
			ϵ/0	**c/1**	**ϵ/0**	$\mathbf{P[A=1,B=1,A_\epsilon,B_c]}$
			c/1	**ϵ/0**	**ϵ/0**	$\mathbf{P[A=1,B=1,A_c,B_\epsilon]}$
			c/1	c/1	c/1	

TABLE I
Ideal AND Gate with Unreliable Inputs

In order to arrive at a close form representation of AND gate output node error probability, we assume (an alternative approach is presented later) that the inputs of the gates are independent. This allows for the use of simple formulas to compute reliability and reduce the overall algorithm running time. The internal node error probability can be expressed as the sum of all the six terms in Tab. I that result in an erroneous output and evaluates to :

$$\begin{aligned} P_\epsilon(\mathbf{Z}^*) = & P_\epsilon(\mathbf{A})P_\epsilon(\mathbf{B})P_\mathbf{A}(0)P_\mathbf{B}(0)+ \\ & P_\epsilon(\mathbf{A})(1-P_\epsilon(\mathbf{B}))P_\mathbf{A}(0)P_\mathbf{B}(1)+ \\ & (1-P_\epsilon(\mathbf{A}))P_\epsilon(\mathbf{B})P_\mathbf{A}(1)P_\mathbf{B}(0)+ \\ & [P_\epsilon(\mathbf{A})+P_\epsilon(\mathbf{B})-P_\epsilon(\mathbf{A})P_\epsilon(\mathbf{B})]P_\mathbf{A}(1)P_\mathbf{B}(1) \end{aligned} \tag{2}$$

Z*		Gate Fault	Z		Error Probability
Ideal	State		Value	State	
0	c	c	0	c	
	c	**f**	**1**	**ϵ**	$\mathbf{P[Z^*=0,Z_c,P_f]}$
	ϵ	**c**	**0**	**ϵ**	$\mathbf{P[Z^*=0,Z_\epsilon,P_c]}$
	ϵ	f	1	c	
1	c	c	1	c	
	c	**f**	**1**	**ϵ**	$\mathbf{P[Z^*=1,Z_c,P_f]}$
	ϵ	**c**	**1**	**ϵ**	$\mathbf{P[Z^*=1,Z_\epsilon,P_c]}$
	ϵ	f	0	c	

TABLE II
Faulty AND Gate with Unreliable Inputs

B. Intrinsic Gate Error Effects

Tab. II presents the analysis of the possible configurations of the faulty buffer. We employ the Binary Symmetric Channel (BSC) model to represent the erroneous buffer behavior. The gate output static and error probabilities can be defined as:

$$\begin{aligned} P_z(1) &= P_z^*(1)(1-P_F) + P_Z^*(0)P_F \\ P_z(0) &= P_z^*(0)(1-P_F) + P_Z^*(1)P_F \\ P_\epsilon(Z) &= P_F + P_\epsilon(Z^*) - 2 * P_F * P_\epsilon(Z^*) \end{aligned} \tag{3}$$

The modeling of unreliable inverter follows similar line of flow. It is not presented here due to space constraints.

Benchmark	Gate Count		Avg Error Deviation on all outputs %			Runtime{s}	
	AND	Inverter	ϵ = 0.001	ϵ = 0.01	ϵ = 0.05	MC_Sims	Tool
cu	55	29	5.75	2.78	9.48	7051.89	**0.393**
x2	55	32	6.06	7.24	9.24	5356.85	**0.924**
parity	45	61	3.51	6.38	7.55	10215.41	**1.042**
cm150a	61	71	3.36	2.81	9.43	16477.93	**1.558**
cordic	82	84	1.57	2.24	9.57	22749.73	**0.966**
mux	85	92	2.32	3.13	6.25	22019.47	**0.295**
b9	104	78	3.79	3.38	6.57	43578.04	**0.827**
count	128	130	5.34	7.84	9.84	52890.57	**4.691**

TABLE III
MCNC BENCHMARK CIRCUITS BASED ACCURACY & PERFORMANCE EVALUATION FOR DIFFERENT GATE ERRORS (ϵ)

IV. CAD TOOL: RELIABILITY EVALUATOR

In this section we present the algorithm and the experimental setup to demonstrate the accuracy of the proposed approach against MonteCarlo simulations. Further, we report the accuracy and simulation time savings compared to MonteCarlo.

A. Computation Algorithm

Alg. 1 presents the methodology employed within the tool to compute circuit reliability. It accounts for the error introduced by both inverter & AND gates. Using Eq. 2 and 3, the error due to the AND gate is computed both on the internal and external output nodes. This flow has been integrated into the open source tool 'abc' [10] and automates the error probability computation.

Algorithm 1 Generic Method for Reliability Evaluation

```
INPUT:N, total number of nodes in the AIG network, Error Probability of Individual Gates and Switching activity $P_{SA}$ on the primary input nodes (PI's)
OUTPUT:Output error probability
  for all nodes I= 1 to N do
    if Input Nodes are inverted then
      Account for the inverter error
    end if
    Compute Internal node error probability using Eq. 2
    Compute Output node error probability using Eq. 3
  end for
```

B. Experimental Flow

Fig. 3 depicts the complete experimental setup developed to compare the algorithm results with fault inserted gate level simulations. The sample size used for reliability analysis was 100k, 50k and 10k for 0.001, 0.01 and 0.05 error scenario's respectively. In Tab. III, columns 1 and 2 give the name and number of gates in the benchmark circuit. Column 3 captures the accuracy of the method when compared with MonteCarlo Simulations while Column 4 highlights the significant time savings the proposed algorithms achieves when compared with MonteCarlo simulations. From the table, it is clear that the proposed algorithm maintains accuracy within 10%. This method while accounting for the impact of reconvergent fanout on static probability does not takes into account the error probabilities. While previous approaches tried to focus on the precise estimation of gate error probability, in the next section, we propose a Bound and Propagate based methodology. To overcome the reconvergent fanout issue, the algorithm computes the upper and lower bounds for the node error probability.

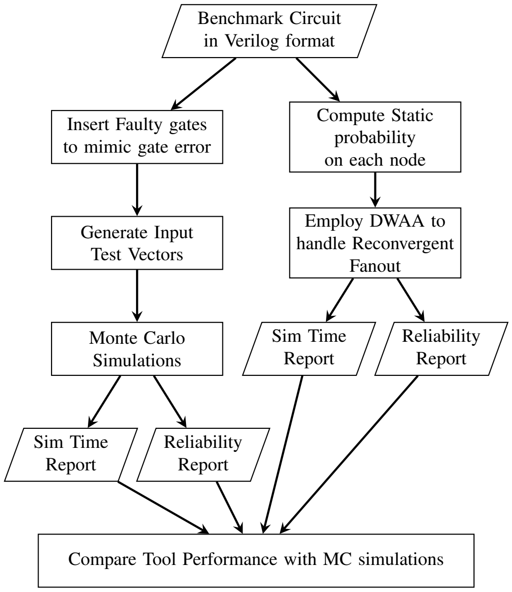

Benchmark Circuit in Verilog format
Insert Faulty gates to mimic gate error
Compute Static probability on each node
Generate Input Test Vectors
Employ DWAA to handle Reconvergent Fanout
Monte Carlo Simulations
Sim Time Report
Reliability Report
Sim Time Report
Reliability Report
Compare Tool Performance with MC simulations

Fig. 3. Design Flow for Reliability Computation

V. RECONVERGENT FANOUT : BOUNDING APPROACH

The methodology developed in the previous section does not accounts for the impact of the reconvergent fanout gate error on the final reliability of the output node. The statistical dependence among signals in a combinational circuit is possible due to reconvergent fanout assuming that the primary inputs are independent. When reconvergent fanout gates are present in a circuit, the signals at the inputs of reconvergent gates are correlated. Ignoring these correlations can produce erroneous results as seen in Tab. III. Computing the node error probability in the presence of reconvergent fanout is complex because Eq. 2 does not hold true. This is because each of the terms in Eq. 2 cannot be factorized due to dependencies between the four probabilities. There is no closed form solution to solve the terms in Eq. 2 in an exact manner. Iterative approaches do exist but their complexity grows exponentially for each of the 6 terms and their running time is not acceptable for application in synthesis tools. This section presents a novel methodology to deal with the impact of reconvergent fanout gate error on the overall circuit error probability.

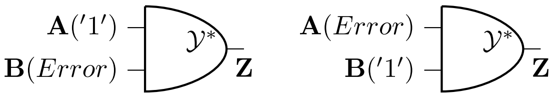


Fig. 4. Bounding Error

A. Bounding Node Error Probability

The methodology of Bound and Propagate is now introduced. It accounts for the reconvergent fanout gate error while not increasing the overall simulation time. The algorithm computes the upper and lower bounds for the error probability on all the gates with the reconvergent fanout. Eq. 2 lists all the possible scenarios that would result in an output error. Bounding each of these terms singularly results in loose bounds that would quickly convergence to the upper/lower bound to 1/0, respectively. To avoid this scenario, only the total gate output error probability is bounded. As depicted in Fig. 4,an error on any of the input nodes can be propagated onto the output iff the other input node is set to '1'. If the other node is at '0', the error would be masked. Now to obtain the bounds, two cases are presented; a pessimistic and an optimistic scenario.

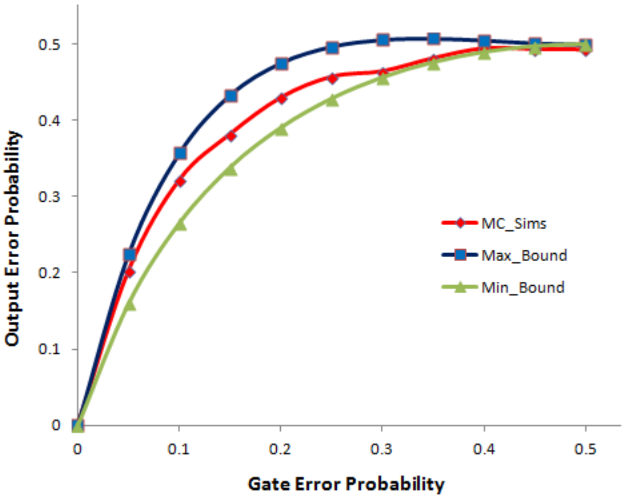


Fig. 5. B9 Benchmark Circuit Error Bounds

Pessimistic Rule: The maximum bound on the error is the summation of the error on both the input nodes. Hence, the sum of the error probabilities on each of the input nodes when the other node is set to '1' provides the maximum possible error on the output of the AND gate. This is represented by Eq. 4. From Tab. I, it is clear that this rule is pessimistic since many cases exist where a single error get "masked".

$$P_\epsilon(Z) \leq P_\epsilon^{Max}(A)P_1(B) + P_\epsilon^{Max}(B)P_1(A) \qquad (4)$$

Optimistic Rule: The guaranteed possible error on the output node of the 'AND' gate is the maximum of the product of the error on the input node and the probability of other node set to '1'.To make the computation more optimistic, we take the effect of only one node into consideration. This is represented by Eq. 5.

$$P_\epsilon(Z) \geq Max\{P_\epsilon(A)P_1(B), P_\epsilon(B)P_1(A)\} \qquad (5)$$

The plot in Fig. 5 illustrate the accuracy of the error bounds when applied to MCNC benchmark circuit 'B9'. We are currently working on extending the concept of error bounds to more complex benchmark circuits.

VI. Conclusion

As part of our ongoing research, we are developing novel reliability aware synthesis algorithms to improve circuit reliability. In this process, reliability evaluation is the most primitive and important step towards developing a reliability aware logic synthesizer. This paper described a simple technique to study the gate failure rate on the overall circuits. The reliability numbers obtained with the proposed algorithm are within 10% margin compared to the MC simulations. This error is within acceptable limits as the main focus of this approach is to quickly compare hundreds of logically equivalent realization and select higher reliability circuit configuration. The next step is to study the impact of reconvergent fanout on reliability estimation in greater detail to develop tighter bounds and eventually integrate it into the synthesis tool.

VII. Acknowledgement

This work was supported by the Seventh Framework Programme of the European Union, under Grant Agreement number 309129 (i-RISC project).

References

[1] S. Borkar. Designing reliable systems from unreliable components: the challenges of transistor variability and degradation. *Micro, IEEE*, 25(6):10–16, 2005.

[2] C. Constantinescu. Trends and challenges in vlsi circuit reliability. *Micro, IEEE*, 23(4):14–19, 2003.

[3] M.R. Choudhury and K. Mohanram. Reliability analysis of logic circuits. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 28(3):392–405, 2009.

[4] Jie Han, Hao Chen, Erin Boykin, and Jos Fortes. Reliability evaluation of logic circuits using probabilistic gate models. *Microelectronics Reliability*, 51(2):468 – 476, 2011.

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

[6] A Biere. The aiger and-inverter graph (aig) format, version 20070427. *Johannes Kepler University*.

[7] Robert Brayton and Alan Mishchenko. Abc: An academic industrial-strength verification tool. In *Computer Aided Verification*, pages 24–40. Springer, 2010.

[8] Rashmi Mehrotra, Tom English, Michel Schellekens, Steve Hollands, and Emanuel Popovici. Timing-driven power optimisation and power-driven timing optimisation of combinational circuits. *Journal of Low Power Electronics*, 7(3):364–380, 2011.

[9] Thiago Figueiro, Renato P Ribas, and André Inácio Reis. Constructive aig optimization considering input weights. In *Quality Electronic Design (ISQED), 2011 12th International Symposium on*, pages 1–8. IEEE, 2011.

[10] A Mishchenko et al. Abc: A system for sequential synthesis and verification. *URL http://www. eecs. berkeley. edu/˜ alanmi/abc*, 2007.