Estimation and Optimization of Reliability of Noisy Digital Circuits

Satish Sivaswamy, Kia Bazargan, Marc Riedel
{satish, kia, mriedel}@umn.edu
Department of Electrical and Computer Engineering
University of Minnesota, MN 55455

***Abstract*— With continued scaling, reliability is emerging as a critical challenge for the designers of digital circuits. The challenge stems in part from the lack of computationally efficient techniques for analyzing and optimizing circuits for reliability. To address this problem, we propose an exact analysis method based on circuit transformations. Also, we propose a hybrid method that combines exact analysis with probabilistic measures to estimate reliability. We use such measures in a rewiring-based optimization framework to optimize reliability. Our hybrid approach offers a speedup of $56X$ when compared to a pure Monte Carlo simulation-based approach with only a 3.5% loss in accuracy. Our optimization framework improves reliability by about 10% accompanied by a 6.9% reduction in circuit area[1].**

***Keywords*— Reliability, Testing, and Fault-Tolerance, Optimization, Automatic Synthesis**

I. Introduction

The continued scaling of silicon-based systems in the deep nanometer regime presents numerous technological challenges. Issues such as thermal fluctuations, quantum effects and radiation strikes manifest themselves as transient errors. These are beginning to affect the functionality and reliability of devices [1], [2], [3]. The impact of transient errors on combinational circuits is projected to be as severe as that on memory elements in the near future [4]. To make matters worse, the failure rates of emerging technologies such as quantum dots and molecular devices are expected to be significantly higher than those of CMOS devices. We will soon be at a point where circuit reliability will be a dominant parameter in the design of circuits. To address this concern, we need fast and accurate techniques for estimating reliability. We need to incorporate these techniques into efficient strategies for optimizing circuits for reliability.

Previous methods that have considered reliability estimation with transient errors are computationally intensive [1], [5], [6]. Some of the work in the area has had a technology-dependent focus, relying on the electrical characteristics of circuit elements [3], [7]. This kind of technology dependence limits the viability for logic-level synthesis. Our contributions in this work are twofold. Firstly, in Section III, we develop efficient reliability estimation techniques using a new signal probability propagation method. Secondly, in Section V, we make use of these techniques to optimize circuit for reliability with an ATPG-based rewiring framework applied during logic synthesis.

II. Fault Modeling

Our aim is to develop logic-level algorithms to estimate and optimize circuit reliability. Accordingly, we adopt the technology-independent fault model used in [1], [2], [5] and [6]. Transient faults are modeled as bit-flips at the outputs of gates and they are assumed to last for exactly 1 clock cycle. The probability with which a gate's output is flipped is its failure probability.

To model a gate G that has a failure probability of ξ, we connect a dummy XOR gate at the output of G. The second input of the dummy XOR gate is connected to a signal, e_i that has a signal probability of ξ. This ensures that the output of G is flipped exactly with probability of ξ. G and the dummy XOR gate together model the faulty gate.

To estimate circuit reliability, first we transform the original circuit by adding the dummy XOR gates mentioned above and then we compare the primary outputs of the transformed faulty circuit to those of the original fault-free circuit. The setup that we use for reliability computation is shown in Figure 1. In this figure, the x_i's form the inputs to the circuit. Each gate g_i has a dummy input ξ_i having a signal probability equal to its failure rate. After transforming the circuit, we compute signal probabilities of the outputs of the fault-free and faulty versions of the circuit. The signal probability of the final dummy XOR gate, f in Figure 1, gives the failure rate and consequently the reliability of the circuit. For multi-output circuits, we have a dummy XOR gate for each output and perform the disjunction of the XOR outputs to compute the overall circuit failure rate. In the next section we present a technique for computing circuit reliability using this setup.

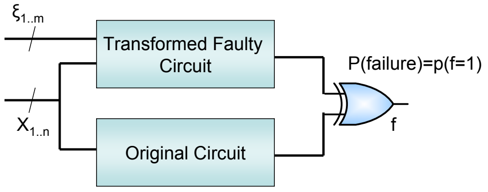


Fig. 1. Setup to Estimate Circuit Reliability

III. Estimating Circuit Reliability

As explained in the previous section, we convert the problem of estimating a circuit's reliability into the problem of calculating the signal probabilities of all of its internal nodes. Many techniques have been proposed to compute signal probabilities in combinational circuits [8]. At one end of the spectrum, there are simulation-based methods which apply large numbers of input vectors. At the other end of the spectrum, there are probabilistic methods that entail propagating probabilities based on simple rules. Probabilistic methods are, in general, more efficient than simulation-based methods. Unfortunately, they suffer from low accuracy due the problem of signal correlations. Several studies have addressed this issue [9], [10], [11]. Although promising, these methods are not accurate enough to be used practically [12]. In [13], the authors propose an exact technique to evaluate signal probability. However, for large circuits their exact method is intractable. [14] partially applies the exact technique to circuits by considering signal reconvergences up to a specified

[1] This work was funded in part by NSF, under grant number CCF-0347891

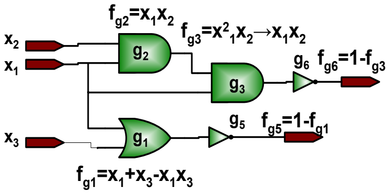


Fig. 2. Illustration of probability propagation

logic depth; beyond this depth, all signals are assumed to be uncorrelated. This results in inaccuracies.

Our ultimate objective is to estimate reliability of nodes in a circuit and use that information in an optimization framework to improve overall circuit reliability. The optimization procedure may involve several iterations where reliability is evaluated to achieve good results. In this context, it is crucial that we we develop fast and accurate techniques to estimate reliability. To this end, we have developed a hybrid approach that combines the exact approach with probabilistic techniques.

A. Preliminaries

To better understand our hybrid approach, in this section we review background information about exact and probabilistic techniques.

1) Exact Approach: The exact method proposed in [13] evaluates signal probability of a node as a polynomial function of the signal probabilities of its inputs. Consider a logic circuit with n inputs, each associated with signal probability variables, $x_1, x_2, \cdots x_n$. Let G be an AND gate with 2 inputs. If $f_1(x_1, \cdots, x_n)$ and $f_2(x_1, \cdots, x_n)$ represent the functions of the signal probabilities of its inputs, the output signal probability is given by the function f_g, such that

$$f_g(x_1, \cdots, x_n) = f_1(x_1, \cdots, x_n) \times f_2(x_1, \cdots, x_n)$$

The corresponding expression for an OR gate is $f_g = f_1 + f_2 - f_1 f_2$ and for an inverter is $f_g = 1 - f_1$. Signal dependencies appear as exponents in these polynomials and are suppressed to achieve accurate results. This is illustrated in Figure 2.

In this figure, x_1, x_2 and x_3 denote the signal probability variables corresponding to the inputs. Note that for gate g_3, application of the probability propagation rules yields a function $x_1^2.x_2$. This indicates that g_3 acts as a site of reconvergence for x_1. The correct function at g_3 is obtained by converting f_{g3} to $x_1 x_2$ by supressing the extra exponent of x_1. In [13], these functions are represented using BDDs and finally the signal probability of all nodes is obtained by performing simulations on these polynomials by assigning Boolean values to $x_1 \cdots, x_n$ according to the input signal probabilities and evaluating the mean values of the polynomials.

In our framework for reliability estimation, shown in Figure 1, in addition to the primary inputs, we also have dummy inputs($\xi_i's$) modeling the failure rates of gates. These can be easily embedded into the probability propagation rules presented in [13]. For example, let g_i be a two-input AND gate with signal probabilities at its inputs given by f_1 and f_2. The signal probability of its output is given by $f_1 f_2(1-\xi_i) + (1 - f_1 f_2)\xi_i$, where ξ_i is the failure probability of the gate.

The ξ_i variables are real numbers and are typically fixed for gates. As a result, unlike [13], we require a convenient way to represent a mapping from a Boolean domain to a real domain ($\Re_+$). So, we can adopt the approach in [5] and use algebraic decision diagrams (ADDs) to represent the polynomial function at each node. Using this method, the evaluation of signal probabilities is exact. Unfortunately, the method is computationally infeasible for large circuits, especially if it is applied in the inner loop of an optimization framework.

2) Probabilistic Approach: To mitigate the cost of the analysis, probabilistic techniques have been widely used to estimate signal probabilities. In this work, we apply the idea proposed in [10] to account for signal dependencies. There the authors introduced the notion of explicit correlation coefficients (interchangeably referred to as correlations in this work) between signals. The correlation coefficient, $C_{i,j}$ between two signals i and j is defined as

$$C_{i,j} = \frac{P(ij)}{P(i)P(j)} = \frac{P(i|j)}{P(i)} = \frac{P(j|i)}{P(j)}$$

Instead of propagating polynomial functions as is done in the exact method, probabilistic techniques work by propagating real numbers corresponding to the signal probabilities of nodes. Probability propagation rules are augmented to include correlation coefficients. For instance, the signal probability at the output of a 2-input AND gate with input signal probabilities given by p_1 and p_2 is given by $p_1 p_2 C_{1,2}$, where $C_{1,2}$ is the correlation coefficient between the two inputs. For an OR gate, the output signal probability becomes $p_1 + p_2 - p_1 p_2 C_{1,2}$.

To propagate correlation coefficients, the authors assume that the circuit's primary inputs are independent of each other. A topological sort algorithm is first used to levelize the circuit. Correlations are computed for each pair of edges that cross a level (i.e. an edge cut-set). This information is used to compute probabilities of nodes in the next level according to the augmented probability propagation rules. This process continues until it reaches the output nodes. Signal correlations between a pair of edges is computed as follows. Let f denote the function implemented by a gate; i and j are its inputs and y is its output. The correlation between y and another edge m can be computed with the help of the following equations [10].

$$\begin{aligned} p(y) &= f(i,j) \\ p(y|m) = p(y)C_{y,m} &= f(i|l, j|l) \\ C_{y,m} &= \frac{f(i|l, j|l)}{f(i,j)} \end{aligned}$$

This method may yield the exact probabilities in certain cases; however, in general, it is approximate since it considers only first-order correlations. Higher-order correlations such as two signals depending simultaneously on a third signal are ignored. In the next section we present a hybrid approach where we combine the features of the exact and probabilistic techniques to develop a fast and accurate reliability estimation algorithm.

IV. Hybrid Approach to Reliability Computation

In this section we develop a hybrid approach for reliability computation that provides the scalability of the probabilistic technique with the higher accuracy of the exact approach. It is clear that if we can partition the circuit into a set of independent regions, we can use the exact approach for smaller regions which may have signal reconvergences inside them and then combine the results using probabilistic techniques.

There are two potential challenges. The first is to identify a set of independent regions in the circuit that are small enough for the exact approach to be applicable. The second

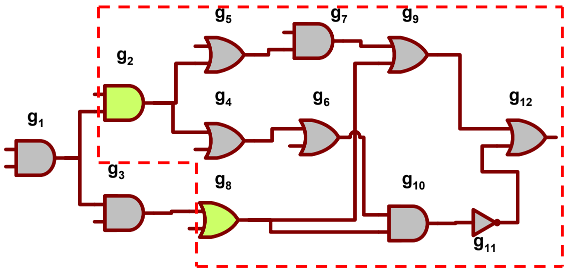


Fig. 3. Illustration of Super-gates

is to develop an interface between the exact and probabilistic techniques to combine the results.

A. Identifying Independent Regions

Previous work such as [15] discussed techniques for partitioning a circuit into a set of independent regions called super-gates. These independent partitions would allow the exact technique to be applied for smaller regions of the circuit. However, from our studies as well as that presented in [14], we note that in real circuits, the sizes of these super-gates are still large. This adversely impacts the applicability of the exact approach. Also, the technique in [15] depends on finding articulation points of the circuit graph. However, after we apply the circuit transformation in Figure 1, the circuit does not have any articulation points.

Ideally, all the sources of reconvergence at a node should be considered exactly in order to accurately evaluate the signal probability of the node. However, this is computationally infeasible. So, we relax this condition and use an approximation where we consider only the *sources of first reconvergence* for every site of reconvergence. The rationale behind this assumption is the observation that the effect of signal dependencies on a node due to reconvergent fanouts decreases as we move away from the node in its transitive fanin-cone. The following definitions are useful in describing the notion of a *source of first reconvergence* for a node.

Definition 1 (Source of Reconvergence): A node u is a source of reconvergence for v if there are at least two pairwise edge disjoint $u \rightsquigarrow v$ paths in the circuit graph

Definition 2 (Sources of first Reconvergence): If S is the set of all sources of reconvergence for u, then the sources of first reconvergence for u is given by the set $S_{first,u}$ defined as

$$S_{first,u} = S - \{v_i | \exists v_i \rightsquigarrow v_j; v_i, v_j \in S; i \neq j\}$$

In our approach, we redefine the notion of a super-gate to represent a maximal pseudo-gate rooted at a site of reconvergence. It has the sources of first reconvergence as its inputs. Note that, unlike the traditional super-gate, the inputs to our super-gate may not be independent of each other. The definitions are illustrated in Figure 3.

Applying Definition 1 to gate g_{12} in the figure, $\{g_1, g_2, g_8\}$ is the set of all sources of reconvergence at g_{12}. However, a path exists from g_1 to g_2 and g_8, so g_1 is removed from this list to form the set $S_{first,12}$. For each super-gate rooted at a reconvergent node, we include only the set of sources of first reconvergence as its input to reduce its size. We treat all gates on the set of pairwise edge disjoint paths from the sources of first reconvergence to the reconvergent node as a super-gate rooted at the reconvergent node. The set $\{g_2, g_4, \cdots, g_{12}\}$ forms the super-gate rooted at g_{12} in Figure 3 (enclosed in a dashed line) and g_2 and g_8 supply its inputs.

We can identify these super-gates by performing depth-first graph traversals from candidate sources in a reverse topological order while keeping track of signal reconvergences.

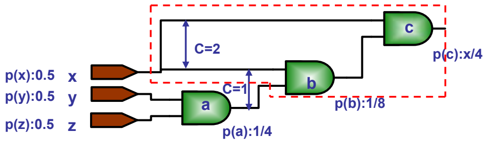


Fig. 4. Evaluating Signal Probability using a Hybrid Approach

B. Combining Symbolic and Probabilistic Techniques

We apply the probabilistic approach for all non-reconvergent nodes and the exact approach for the super-gates rooted at reconvergent nodes. The treatment of non-reconvergent nodes is the same as in Section III-A.2. The exact technique cannot be directly applied to super-gates for evaluating the signal probabilities of reconvergent nodes. The rest of the section describes how to adapt the exact and probabilistic techniques for this.

To capture the effects of signal correlation from the sources of first reconvergence for a super-gate, we assign a Boolean variable to each source of reconvergence. This is similar to the treatment of primary input nodes in the exact approach. The difference in the hybrid approach is that we may have gates other than sources of first reconvergence that supply inputs to some gates inside the super-gate. We treat these inputs as pseudo-inputs. Instead of assigning separate Boolean variables for these inputs, we treat them as real numbers representing probability values. This minimizes the size of the decision diagram that is needed to represent the polynomial function representing the signal probability at the root of the super-gate.

Note that in the exact approach, the coefficients of each term in the polynomial function is always either 1,-1 or 0. (See 2.) However, this is no longer true in our work since we consider the pseudo-inputs as real numbers. The coefficients for the polynomial representing the signal probability of the super-gate root may take any value.

Since we treat the pseudo-inputs as real numbers, we may lose some correlation information if they are correlated to some of the sources of first re-convergence at an upstream node. To capture this effect, we embed the correlation between pseudo-inputs and signals present in the super-gate while building the decision diagram of the super-gate. Correlation information is stored as a part of the function being evaluated at the super-gate root. This function is stored as an ADD.

To compute the signal probability of the root node of a super-gate, we generate Boolean vectors for the sources of first reconvergence based on their probabilities (these are already known since we proceed in a topological order) and evaluate the decision diagram of the root node. The mean of the decision diagram values at the root node gives its probability.

We use the example in Figure 4 to illustrate the process of building the ADD of a super-gate, embedding correlation information and finding probabilities of reconvergent nodes. Nodes x, b and c constitute the super-gate rooted at c. Node x is a source of first reconvergence and is given a separate Boolean variable in the decision diagram. Here a is a pseudo-input and is treated as a real number. $C_{x,a}$ (=1 in the example since x and a are independent) is the correlation between the pseudo-input and a source of first reconvergence. Assuming the gates are fault-free, the probability of c is given by the function,

$$p(c) = xp(a)C_{x,a} = x.\frac{1}{4}.1 = \frac{x}{4}$$

We store this function as an ADD at node c. To evaluate this function, we apply Boolean values 0 and 1 to x with a probability of 0.5 each since $p(x) = 0.5$ and evaluate the decision diagram at $c(p(c))$. This gives the correct probability of c as 1/8.

Evaluating the probability of the root node of a super-gate this way will lead to inaccuracies if the sources of first reconvergence themselves are correlated (gates g_2 and g_8 for instance in Figure 3). We have developed a heuristic to correct this problem.

Suppose there are k sources of first reconvergence for a super-gate. Let $n_1 \cdots n_k$ be the k unique symbolic Boolean variables we assign them. We order these nodes according to their levels in the circuit graph. When we evaluate the probability of the super-gate root, we start from n_1 and assign it either 0 or 1 based on its probability. If we have assigned values to m nodes, the probability for the $(m+1)^{st}$ node is adjusted as the conditional probability based on the previous m assignments as follows.

$$\begin{aligned} p_{adjusted}(m+1) \quad &= \quad p(n_{m+1}|(n_1 = b_1, \cdots, n_m = b_m) \\ &\quad\quad , b_i \in \{0,1\}) \qquad (1) \end{aligned}$$

If e denotes the event that $n_1 = b_1$,$n_2 = b_2$,$\cdots$ $n_m = b_m$, (1) reduces to

$$p_{adjusted}(m+1) = p(n_{m+1}).C_{n_{m+1},e} \qquad (2)$$

We consider only first order correlations between sources of first reconvergence. So, we approximate the correlation term in (2) as

$$C_{n_{m+1},e} \approx C_{n_{m+1},n_1=b_1} \times \cdots \times C_{n_{m+1},n_m=b_m}$$

$C_{n_{m+1},n_i=1}$ is computed the same way as in Section III-A.2. $C_{n_{m+1},n_i=0}$ is given by

$$C_{n_{m+1},n_i=0} = p(n_{m+1}).\frac{(1 - p(n_i)C_{n_{m+1},n_i=1})}{(1 - p(n_i))} \qquad (3)$$

We generate Boolean vectors for the sources of first reconvergence based on these adjusted probabilities and then evaluate the probability of the super-gate root by using the method described earlier in this section.

The overall circuit reliability is obtained by applying the probabilistic approach for all non-reconvergent nodes and by applying the technique described in this section for reconvergent nodes in the circuit.

In the next section we describe our experimental setup to validate our work and compare our approach with the exact and probabilistic techniques described in this section.

C. Validating the Hybrid Approach

To validate the accuracy and efficiency of our hybrid approach, we implemented reliability estimation techniques based on the exact, probabilistic and hybrid signal probability evaluation approaches and compared them to a Monte Carlo-based fault injection framework. For each input vector sample in the Monte Carlo simulation, we can perform fault simulations and inject faults at gates based on their failure probabilities. We assume that the primary inputs are all independent[2] with a probability of 0.5 and failure rate of gates, ξ is 0.005. In our symbolic and hybrid reliability computation techniques, we sample the input space of the function being evaluated until the final probability converges.

We implemented all techniques presented in this work in C++ and used benchmark circuits from the LGSynth93 benchmark suite. The results are presented in Table I. Exact, Prob, Hyb and Sim refer to exact, probabilistic, hybrid and Monte Carlo approaches to reliability estimation, respectively. From the results, it is clear that the exact approach works well for small circuits but is infeasible for larger circuits due to the exponential increase in the size of the decision diagrams involved. So, the exact approach cannot be used in an optimization framework. We note that the sizes of circuits that we have used are larger than those demonstrated in [6] and [5].

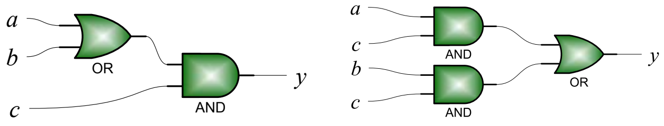
The errors of the probabilistic and hybrid approaches are about 6.4% and 3.5% on average, respectively, when compared with results from Monte Carlo-based fault injection. The maximum error of the hybrid approach is 8.31% as opposed to 17.16% for the probabilistic approach indicating that the hybrid approach is more consistent overall. The errors of both techniques increase with the size of circuits. For the 4 largest circuits in Table I, the average errors are 10.4% for the probabilistic approach and 4.2% for the hybrid approach. An increase in the estimation error for large circuits is intuitive because of a greater likelihood of signal reconvergences in the circuit. This may lead to more approximations in the computation. Both techniques still perform reasonably well.

From Table I, it is seen that on average, the exact technique performs very well for small circuits since the decision diagrams involved are smaller and evaluating them is easier than propagating correlations across edge cut-sets. Over all circuits used, the probabilistic approach was the most efficient. This is expected since our hybrid approach involves all the computation involved in the probabilistic approach in addition to identifying and evaluating super-gates. When compared to the Monte Carlo approach, the probabilistic and hybrid approaches offer average speedups of $131X$ and $56X$, respectively.

In the next section, we present a rewiring-based reliability optimization framework based on our hybrid reliability evaluation technique.

V. Rewiring-based Reliability optimization

Reliability can be optimized by selectively hardening gates that are sensitive to transient errors by up sizing them. We observe that the topology of circuits also has a significant impact on reliability. This is demonstrated in Figure 5, which shows two circuit realizations of the same logic function. The difference in their reliabilities is about 5%.



(a) Circuit A, Reliability : 0.927 (b) Circuit B, Reliability : 0.884

Fig. 5. Reliability for different circuit topologies

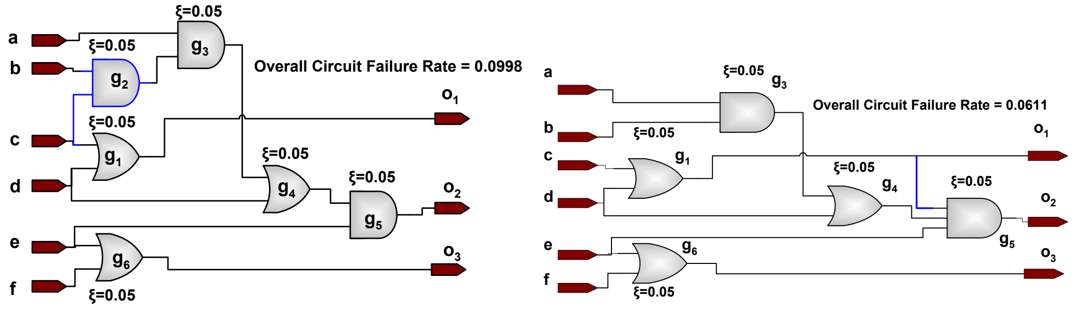
The potential of rewiring-based approaches to optimize reliability is illustrated in Figure 6. The wire $c \Rightarrow g_2$ is replaced by $g_1 \Rightarrow g_5$ after rewiring. This makes g_2 redundant; it is removed from the circuit. We observe that the failure rate of the circuit falls from 0.0998 to 0.0611 after the rewiring transformation. There are two main reasons for this. Firstly, rewiring reduces the number of gates through which c passes through before reaching O_2. This reduces the chances of errors in the path. Secondly, as a by-product of rewiring, we

[2] We can easily extend our work to include the case where inputs are correlated

TABLE I

Results of Reliability Estimation

Circuit	Reliability				Runtime (sec)				Error(%)		
	Exact	Prob	Hyb	Sim	Exact	Prob	Hyb	Sim	Prob	Hyb	Sim
fulladder	0.7451	0.7642	0.7332	0.7482	0.1	0.1	0.1	18.98	2.56	1.59	0.42
C17	0.8315	0.8577	0.8519	0.8251	0.1	0.1	0.2	17.54	3.15	2.44	0.76
b1	0.8057	0.7988	0.8015	0.1	0.1	0.1	18.61	0.14	0.99	0.66	
cm42a	0.9085	0.8584	0.8913	0.9094	0.1	0.2	0.3	48.22	5.51	1.89	0.09
cm138a	0.9114	0.9078	0.9078	0.9152	0.1	0.4	0.2	52.23	0.39	0.39	0.41
tcon	0.8237	0.8561	0.7985	0.8312	0.1	0.5	0.5	56.61	3.92	3.06	0.90
count	0.6840	0.7801	0.7152	0.6903	0.4	0.77	1.49	317.33	14.05	4.56	0.93
c8	0.7601	0.7626	0.6969	0.7527	0.2	0.83	1.68	361.98	0.33	8.31	0.96
sqrt	0.5287	0.5881	0.5542	0.5295	0.2	0.48	1.86	343.79	11.21	4.82	0.14
term1	0.7392	0.7547	0.7429	0.7342	8.06	5.38	10.82	949.02	2.09	0.50	0.68
alu4	0.4436	0.4786	0.4498	0.4494	16.14	14.06	44.76	1589.01	7.88	1.38	1.29
C432	x	0.3543	0.3835	0.4136	x	4.11	8.23	745.43	14.33	7.27	x
too_large	x	0.1861	0.1709	0.1588	x	22.76	45.16	1994.92	17.16	7.59	x
Mean					**2.33**	**3.83**	**8.87**	**501.05**	**6.37**	**3.45**	



a
b
c
d
e
f
ξ=0.05
g1
g2
g3
g4
g5
g6
Overall Circuit Failure Rate = 0.0998
Overall Circuit Failure Rate = 0.0611
o1
o2
o3

(a) Original Circuit

(b) Circuit after Rewiring

Fig. 6. Reliability for different circuit topologies

were able to reduce the number of gates in the circuit. This automatically reduces the failure rate. Additionally, the area that is saved by rewiring can be used to further reduce the failure rate by selective gate up sizing.

A. Rewiring Framework

We adopt an ATPG based rewiring approach for incrementally restructuring a circuit to improve its reliability. The rewiring engine first identifies a set of mandatory assignments to nodes to propagate a signal on the wire to be replaced to the outputs. Then, a set of candidate wires which, when added to the circuit, would make the target wire redundant are identified. Finally, the target wire is replaced by one of the candidate wires. The ability to rewire signals as well as the choice of available candidate connections depend on the number of mandatory assignments that can be identified.

Our rewiring engine is similar to [16], which uses direct implication to uncover mandatory assignments. To uncover more mandatory assignments we provide a parameterized backtracking procedure where we recursively perform dynamic implication to a few logic levels as specified by the user. Implicants uncovered this way are verified by casting the circuit as an instance of a SAT problem and checking the implicants uncovered by backtracking for consistency. We also use the blocking clauses generated by the SAT solver to uncover more implicants. This is again controlled by a user parameter in order to trade off runtime with the number of implicants uncovered.

We can optimize reliability by first identifying nodes in the circuit that have low reliability. We can then rewire signals connecting to these nodes by replacing them with alternate wires driven by more reliable sources in the circuit. If we select the nodes and thereby the wires to replace intelligently, the overall circuit reliability can be improved. We have developed cost metrics to help us identify target wires to be replaced as well as their alternate wires. We explain these metrics in Section V-A.1.

1) Cost metrics for rewiring: Our approach is based on identifying nodes with low reliability. We modify the reliability estimation setup shown in Figure 1 to insert dummy XOR gates for every node in the original circuit to obtain the reliability of each node in addition to the overall reliability. We also use the concept of *observability* of a node, which is a measure of the likelihood that the logic value at the node is visible at the output. We use the approach in [10] to evaluate the observabilities of all nodes in the circuit as probabilistic measures.

We use the reliabilities and observabilities of internal nodes in the circuit to select a target node according to the following cost measure:

$$C_t(N) = (1-\alpha_t-\beta_t).(p_{fail}(N))+\alpha_t.obser.(N)+\beta_t.\frac{l_N}{L_{max}}$$

where C_t is the cost of the target node, $p_{fail}(N)$ is its failure probability, $observ.(N)$ is its observability, l_N is its level in the circuit and L_{max} is the maximum depth of the circuit. α_t and β_t are parameters that control the relative importance of these three factors and are tuned experimentally. The terms containing observability and level of the node give importance to nodes that are highly visible and close to the output. This is important to ensure that significant effort is not wasted on nodes that may have poor reliabilities but whose errors are

TABLE II

RESULTS OF REWIRING OPTIMIZATION

Circuit	Initial Rel	Area	Final Rel.	Final Area	Rel Impr	Area Impr
sqrt8	0.5293	1068	0.5589	1010	5.6	5.43
C432	0.4139	1420	0.5549	1248	34.08	12.11
alu4	0.4492	4554	0.5054	448	12.53	2.33
term1	0.7336	2966	0.7972	2392	8.68	19.35
too_large	0.1567	6078	0.1670	5962	6.58	1.91
count	0.6983	858	0.8095	678	15.93	20.98
tcon	0.8352	210	0.9334	194	11.76	7.62
cm42a	0.8997	140	0.9482	132	5.4	5.71
Mean					10.41	6.9

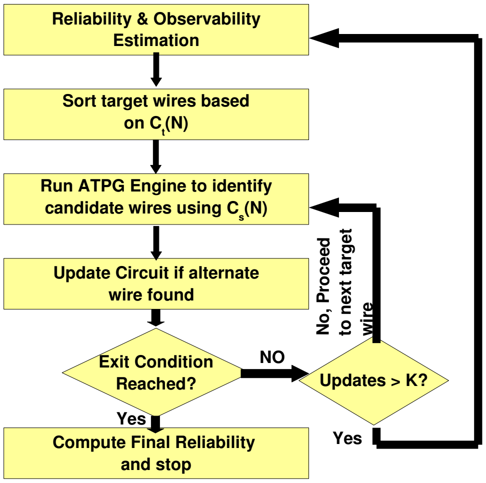

Reliability & Observability Estimation
Sort target wires based on $C_t(N)$
Run ATPG Engine to identify candidate wires using $C_s(N)$
Update Circuit if alternate wire found
Exit Condition Reached?
NO
Updates > K?
No, Proceed to next target wire
Yes
Compute Final Reliability and stop
Yes

Fig. 7. Rewiring flow

unlikely to propagate to the outputs. We select the target node based on this cost measure and try to rewire the inputs to this node in the decreasing order of their failure rates.

We select the source of the candidate wires based on the following cost measure.

$$C_s(N) = (1 - \beta_s).(p_{rel}(N)) + \beta_s(1 - \frac{l_n}{L_{max}})$$

Where $p_{rel}(N)$ is the reliability of gate N, i.e. $1 - p_{fail}(N)$. The other terms have a similar meaning to the target node described above. We select the source nodes of the candidate wires to have high reliability and close to the inputs. This will reduce the chances of an error on the source of the candidate wire. We give preference to nodes that are closer to the output when selecting the destination node of the candidate wire.

After selecting the target and candidate wires, we replace the target wire by the candidate wire and update the circuit. For performance reasons, we re-evaluate circuit reliabilities and observabilities after every k updates. We repeat this process until either there are no more rewiring options or until the overall reliability stops improving for a few iterations. Our rewiring flow is shown in Figure 7.

B. Rewiring Results

The results of our rewiring-based reliability optimization framework is presented in Table II. The experimental setup is similar to the one presented in Section IV-C. A Monte Carlo simulation-based reliability estimation approach is used to evaluate the pre- and post-optimization reliabilities listed in the table under column *Initial Rel* and *Final Rel*, respectively. Our hybrid reliability estimation approach is used in the rewiring framework shown in Figure 7. The initial and final circuit area is reported in columns *Area* and *Final Area*, respectively. We assume that all gates are made of minimum width transistors and report the transistor count as the area measure.

Our rewiring based optimization improves overall circuit reliability by about 10% on average with a 6.9% improvement in area. This demonstrates that rewiring is an effective strategy for optimizing circuit reliability. Furthermore, we obtain improvements in both area and reliability indicating runtime is the only penalty of this approach. The savings in area can be used to further improve reliability by selectively upsizing gates that are more sensitive to transient errors.

REFERENCES

[1] J. Han, E. Taylor, J. Gao, and J. Fortes, “Faults, error bounds and reliability of nanoelectronic circuits,” in *International Conference on Application-Specific Systems, Architecture and Processors*, 2005.

[2] J. P. Hayes, I. Polian, and B. Becker, “A model for transient faults in logic circuits,” in *International Workshop on Design and Test*, 2006.

[3] N. Miskov-Zhivanov and D. Marculescu, “Mars-c: Modeling and reduction of soft errors in combinational circuits,” in *Proc. DAC*, 2006.

[4] P. Shivakumar, S. Keckler, D. Burger, M. Kistler, and L. Alvisi, “Modeling the effect of technology trends on the soft error rate of combinational logic,” in *Proc. Intl. Conf. on Dependable Systems and Networks*, 2002.

[5] S. Krishnaswamy, G. F. Viamontes, I. Markov, and J. P. Hayes, “Accurate reliability evaluation and enhancement via probabilistic transfer matrices,” in *Design Automation and Test in Europe*, 2005.

[6] D. Bhaduri and S. Shukla, “Nanoprism: A tool for evaluating granularity vs. reliability trade-offs in nano-architectures,” in *Great Lakes VLSI symposium*, April 2004, pp. 109–112.

[7] C. Zhao, S. Dey, and X. Bai, “Soft-spot analysis: targeting compound noise effects in nanometer circuits,” *IEEE Design and Test*, vol. 22, pp. 362–375, 2005.

[8] F. Najm, “A survey of power estimation techniques in vlsi circuits,” *IEEE. Transactions on VLSI systems*, vol. 2, no. 4, pp. 446–455, 1994.

[9] B. Krishnamurthy and I. G. Tollis, “Improved techniques for estimating signal probabilities,” *IEEE. Transactions on Computers*, vol. 38, no. 7, pp. 1041–1045, 1989.

[10] S. Ercolani, M. Favalli, M. Damiani, P. Olivo, and B. Riccó, “Testability measures in pseudorandom testing,” *IEEE Transactions om Computer-Aided Design*, vol. 11, no. 6, pp. 794–800, 1992.

[11] B. Kapoor, “Improving the accuracy of circuit activity measurement,” in *Proc. DAC*, 1994.

[12] R. Marculescu, D. Marculescu, and M. Pedram, “Efficient power estimation for highly correlated input streams,” in *Proc. DAC*, 1995.

[13] K. Parker and E. J. McCluskey, “Probabilistic treatment of general combinational networks,” *IEEE. Transactions on Computers*, vol. c-24, 1975.

[14] J. Costa, L. Silveira, and S. Devadas, “Power estimation using probability polynomials,” *Design Automation for Embedded systems*, vol. 9, pp. 19–52, 2004.

[15] H. Min and E. Park, “Graph-theoretic algorithm for finding maximal supergates in combinational logic circuits,” *IEE Proceedings-Circuits, Devices, Systems*, vol. 143, no. 6, pp. 313–318, 1996.

[16] S. Chang, L. van Ginneken, and M. Sadowska, “Circuit optimization by rewiring,” *IEEE Transactions on computers*, vol. 48, no. 9, pp. 962–970, 1999.