Reliability-driven don't care assignment for logic synthesis

Andrew Zukoski Mihir R. Choudhury Kartik Mohanram Department of Electrical and Computer Engineering, Rice University, Houston zukoski@rice.edu mihir@rice.edu kmram@rice.edu

Abstract

This paper describes two algorithms for the selective assignment of input don't cares (DCs) for logical derating of input errors to enhance reliability. It is motivated by the observation that reliabilitydriven assignment of DCs can improve input error resilience by up to 49.7% in logic circuits. Two algorithms - ranking-based and complexity-factor-based - for reliability-driven DC assignment are proposed in this paper. Both algorithms use Hamming distance metrics to determine 0/1 assignments for the most critical DC terms, thereby leaving flexibility in the circuit specification for subsequent optimization. Since ranking-based DC assignment offers less control over overhead, we develop a complexity-factorbased DC assignment algorithm that can achieve up to 21.4% improvement in error rate with a simultaneous 4.3% reduction in area over conventional DC assignment. Finally, we derive analytical estimates on min-max reliability improvements to evaluate the effectiveness of the proposed algorithms.

1. Introduction

The traditional hardware design paradigm has relied on a combination of (i) design for the worst case and (ii) rigorous manufacturing test, diagnosis, and debug to achieve high process yield and product quality. However, increased process variations, wearout failure mechanisms, as well as susceptibility to single event upsets are projected to alter this paradigm in the near future, e.g., [1, 2]. Motivated by these concerns, reliability has emerged as an important design metric, considered on an equal footing with performance and power in integrated circuit design. Whereas several techniques to enhance reliability — spanning devices to architectures — have been proposed in literature (e.g., [3–5]), relatively few are cost effective for use in multi-level logic circuits. Furthermore, these techniques focus on mitigating the effects of errors due to gate and/or wire/fanout errors within the circuit, and not on improving the resilience of the circuit to errors at its inputs.

This paper describes two algorithms for the selective assignment of input don't cares (DCs henceforth) for logical derating of input errors. Since error derating is critical in determining system-level failure rate [6], the proposed technique can effectively contribute to improvements in system reliability. To the best of our knowledge, this is the first work that explicitly focuses on increasing error derating at the technology-independent stage of logic synthesis. Since DC assignments are completely determined at the functional level, the proposed reliability-driven DC assignment algorithms can be integrated into a conventional logic optimization flow. This work is distinct from [7], where DCs are used during LUT mapping of a design to reduce error propagation in FPGAs with faulty configuration bits.

The algorithms for DC assignment described in this paper use Hamming distance metrics to determine selective 0/1 assignments for the DCs in the function specification. If an input error causes

978-3-9810801-7-9/DATE11/©2011 EDAA

the output to shift from the off-set to the on-set, or vice versa, in the specification, the error will propagate to the outputs. However, if this shift does not occur, the error will be masked. In reliability-driven assignment, DCs are assigned to 0/1 to minimize these shifts, thereby increasing the logical error derating capability of the specification to input errors. In the simplest implementation, the DC minterms are assigned in decreasing order of the reductions in shifts they provide. This ranking-based DC assignment algorithm assigns only the DCs most critical for reliability, leaving flexibility in the specification for subsequent optimization.

Sweeping between complete reliability-driven ranking-based assignment and complete conventional assignment of DCs, we observe a tradeoff of performance for reliability as more DCs are assigned for reliability. We also observe that selective ranking-based DC assignment can result in a simultaneous improvement in circuit performance and area up to 24.5% and 26.4%, respectively. These assignments avoid the high overheads associated with assigning all DCs for reliability, when improvements of 49.7% are possible for area overhead in excess of 100%. To better understand reliabilityoverhead tradeoffs, we explored ranking-based DC assignments on synthetic benchmarks of varying Boolean complexity factors. The complexity factor [8] of a Boolean function is an accurate predictor of the size of the minimal sum-of-products (SOP). When rankingbased DC assignment was applied on these synthetic benchmarks, we observed that functions with low complexity factor (i.e., functions with large SOPs) are most amenable to reliability-driven DC assignment, showing simultaneous performance and reliability improvements of 20% and 17% on average, respectively.

With this knowledge, we propose a complexity-factor-based assignment algorithm for DCs that is based on an extracted "local complexity factor". The local complexity factor guides the algorithm to avoid DCs that, if assigned for reliability, can potentially result in high performance overheads. We report results on both synthetic and published benchmarks, demonstrating that the complexity-factor-based algorithm is capable of greater reliability improvements without area overhead, in comparison to the rankingbased assignment. Simulation results show that complexity-factorbased DC assignment is capable of as much as a 21.4% improvement in reliability with a simultaneous 4.3% improvement in area. Simulation results were obtained using Synopsys Design Compiler, and qualitatively validated by the academic synthesis tool ABC [9].

Finally, we derive analytical estimates for min-max reliability improvements to evaluate the effectiveness of the proposed DC assignment algorithms. In the first approach, the analytical estimates are derived using the fraction of minterms in the off-, on-, and DCsets of a given function. The second approach improves the accuracy by expressing the estimates using 1-Hamming distance neighbor statistics of the given function.

This paper is organized as follows. Section 2 motivates error derating using reliability-driven DC assignment. Section 3 describes the ranking-based DC assignment. Section 4 describes the complexity-factor-based DC assignment. Section 5 derives analytical estimates for reliability improvements. Section 6 is a conclusion.

This research was supported in part by NSF CAREER Award CCF-0746850 and in part by NSF grant CCF-0917226.

2. Background and motivation

Of the many solutions proposed in literature to increase circuit reliability to hardware failures, there has been a recent surge in the development of low-cost solutions based on concurrent error detection and concurrent error masking [10–15]. This is because these solutions have the capability to enhance reliability to a broad class of temporary and permanent faults, and require low support from the higher layers for operation.

Unlike traditional reliability-driven circuit synthesis, where errors arising from gate and/or wire/fanout failures within a logic circuit have been the focus of optimization, this paper focuses on input errors due to propagated failures. In other words, this paper focuses on failures propagated from previous blocks and latched as input errors to the block in question. Unlike errors arising due to failures within the circuit, the masking of input errors to improve reliability is independent of the structural implementation of the circuit: input errors are not transient, and the propagation of an input error depends only on the relative phase of the function specification for the correct and erroneous input vectors. However, the masking rate of input errors is dependent on the assignment of DCs in the function specification. For instance, if the correct input vector to a logic circuit is 0100 and the third input to the circuit fails, the input vector 0110 will be applied to the logic circuit. If the function implementation is identical for these two vectors, then the error will be masked. If 0110 is in the don't care (DC) space of the function specification, then assignment of this minterm to either 0 or 1 will determine the masking of an error on the third input with input vector 0100. This example demonstrates that DC assignment will influence the aggregate rate at which input errors are masked.

Reliability-driven DC assignment of function specifications has not been explored in literature. The DC space offers wide flexibility, but has traditionally only been used to minimize the objectives of area, delay, and power. In this paper, we propose an algorithm for DC assignment that achieves both reliability and area improvements over traditional optimization algorithms.

Throughout this paper, we compare reliability-driven DC assignment to conventional DC assignment using Synopsys Design Compiler. We report area, delay, and power consumption while optimizing for delay and power in turn. At all times, reliability is reported in terms of resilience to single-bit input errors: numerical values indicate the fraction of input errors that propagate to each output. Assuming that errors on different input pins are uncorrelated, and that errors are relatively infrequent, the relative occurrence of single-bit errors will far exceed that of multi-bit errors.

2.1 Motivating example

To motivate how reliability-driven DC assignment differs from conventional assignment, consider the incompletely specified function in Figure 1. Conventionally, DC minterm assignments to 0/1 are made to minimize the SOP covering the function. In this example, all three minterms are assigned to logical 1.

Examining these minterms in turn, consider assignment for improving reliability — interpreting reliability as the resilience of the function to single-bit input errors. Assigning DC minterms so as to minimize the number of pairs of neighboring (1-Hammingdistant) minterms with opposite phase will also minimize the number of output transitions resulting from single-bit input errors, and thereby maximize circuit reliability under our interpretation. The first DC (x_1) has two neighboring on-set minterms, and a single off-set neighbor. Hence, assignment of x_1 to the on-set will mask two of the three possible single-bit input errors while assignment to the off-set will only mask one. The fourth uncounted input that can shift the output to x_1 originates in the DC-set (at x_2), which



Figure 1: Reliability-driven vs. conventional DC assignment

can never occur in practice. Hence, x_1 is assigned to the on-set. The second DC (x_2) has two off-set neighbors and a single on-set neighbor; hence, it is assigned to the off-set. The third DC (x_3) has two neighbors each in the on- and off-sets. As a result, assignment to either the off-set or the on-set would mask two single bit input errors, and the minterm is left unassigned.

This example demonstrates that reliability-driven assignment can be in direct agreement or conflict with conventional assignment, or remain ambiguous, retaining functional flexibility for later performance optimization. In practice, we found that reliability-driven DC assignment typically conflicted with conventional DC assignment for around 30% of minterms.

2.2 Synthetic circuits and complexity factor

The number of published benchmarks with explicitly defined input DC sets is small, limiting testing of reliability-driven assignment. Whereas DC sets can be extracted from benchmarks with completely specified functionality [16], this requires considerable effort and is not the focus of this paper. To increase the pool of available benchmarks, we have developed synthetic benchmarks that allow us to explore a range of circuit functionalities. Completely random benchmarks can be generated directly by flipping a three-sided coin for each minterm. However, this results in homogeneous functions that bear little resemblance to published benchmarks, which contain more structure and result in smaller implementations for similar signal probabilities and input counts. To account for this, we generate synthetic benchmarks with a designated complexity factor as follows.

The complexity factor for an *n*-input function *f* is defined in literature [8] as the number of minterm pairs (x_1, x_2) such that (i) x_1 and x_2 share the same phase (on, off, or DC) and (ii) the Hamming distance $D_H(x_1, x_2) = 1$. We normalize the complexity factor to aid comparison between functions with differing pin counts. The normalization factor is equal to the total number of ordered pairs of neighboring minterms (*n* neighbors, 2^n minterms)

$$C^{f} = \frac{1}{n2^{n}} \left| \left\{ (x_{1}, x_{2}) \middle| f(x_{1}) = f(x_{2}) \text{ and } D_{\mathrm{H}}(x_{1}, x_{2}) = 1 \right\} \right|$$

The complexity factor is an accurate predictor of the size of the function's minimal SOP representation [17]. The normalized com-



Figure 2: SOP size vs. complexity factor for ten input, single output synthetic benchmarks

plexity factor represents the probability that a given neighbor of a minterm will be in the same phase. We computed it by iteratively examining neighboring minterm pairs. At the extremities, a normalized complexity factor of 1 indicates a constant function. A normalized complexity factor of 0 (for a fully specified function) represents a perfect XOR. This definition is at first counterintuitive: a high complexity factor implies a 'simpler' function, but this is the definition previously used in literature.

Figure 2 shows the number of implicants in a minimal SOP representation (generated by ESPRESSO) as a function of complexity factor for 10 input, single output functions. As expected, the implicant count converges to 512 with low complexity factor, and declines smoothly to 0 as the complexity factor increases. This demonstrates the ability of our approach to generate a wide range of functionalities necessary to evaluate the effectiveness of the algorithms for DC assignment proposed in this paper. Table 1 below shows the complexity factors for the benchmarks used in this paper.

Table 1: Published and synthetic benchmark properties

Be	nchma	rk	%DC	Complexity Factors			
Name	Inputs	Outputs	<i>n</i> DC	$\mathbb{E}[C^f]$	C^{f}		
bench	6	8	68.9	.533	.540		
fout	6	10	41.4	.351	.338		
p3	8	14	79.6	.671	.805		
p1	8	18	77.7	.641	.788		
exp	8	18	77.2	.644	.788		
test4	8	30	71.5	.560	.557		
ex1010	10	10	70.3	.540	.539		
exam	10	10	86.8	.768	.802		
t4	12	8	43.9	.477	.867		
random1	12	12	68.6	.52	.49		
random2	12	12	68.6	.52	.667		
random3	12	12	68.6	.52	.826		

3. Ranking-based DC assignment

To evaluate reliability-driven assignment, we chose several benchmarks from the MCNC set that have explicitly defined DC spaces. These benchmarks were originally specified in the .pla format. For ranking-based DC assignment, functions were passed into a tool implementing the algorithm outlined in Figure 3, where the onset, off-set, and DC-set are independently maintained and manipulated using the CUDD BDD package [18]. This algorithm ranks minterms based on their possible contribution to circuit reliability — the difference between the number of on-set and off-set neighbors. A given fraction of the DCs are then assigned in this order.

```
inputs: function specification, fraction (of DCs to be assigned)

foreach DC minterm x_i

w_i = ||on-neighbors(x_i) - off-neighbors(x_i)||

if (w_i \neq 0) DC_List.insert(x_i)

sort DC_List in the decreasing order of w_i

for i = 0 to (fraction × DC_List.length)

if (on-neighbors(x_i) > off-neighbors(x_i)) x_i \leftarrow 1

else x_i \leftarrow 0

for i = (\text{fraction} \times \text{DC}_\text{List.length} + 1) to DC_list.length

Leave x_i as DC
```

Figure 3: Proposed ranking-based DC assignment

After ranking-based DC assignment, the function specifications were passed as .pla files for assignment of remaining DCs, synthesis, and mapping to a 70nm standard cell library using Synopsys



Figure 4: Normalized error rate as a function of the fraction of DCs assigned for reliability.

Design Compiler. Within Synopsys Design Compiler, circuits were optimized for delay and power in turn. The options "set_max_delay -to [all_outputs] 0" and "set_max_leakage_power 0; set_max_dynamic_power 0" were used, respectively.

Figure 4 reports the normalized error rate of the specification, as a function of the fraction of DCs assigned by the ranking-based algorithm. The error rate is normalized by the error rate for complete conventional DC assignment (0% ranking-based DC assignment). The results in Figure 4 show that reliability-driven assignment is effective at increasing the resilience to input errors, and that circuit resilience increases as more DCs are assigned for reliability.

Figure 5 reports the effects of ranking-based DC assignments on overhead. We observed that optimization for minimum area (using "compile -area_effort high") resulted in implementations very similar to the power-optimized cases. Hence, for brevity, we report only the results for power and delay optimization. In both cases, normalized area, delay, and power are reported in Figure 5. As expected, there is a tradeoff between reliability and overhead — the mean values for area, delay, and power increase with the fraction of assignments made for reliability. However, in some cases, rankingbased assignment results in implementations that are both smaller and more robust (consider the points of the minimum area line that lie below 1.0). This is unexpected, as ranking-based assignment at no point explicitly optimizes for performance, and is often in conflict with the conventional DC assignments of Synopsys Design Compiler. Thus, ranking-based assignment guides later, conventional optimizations to more optimal configurations. To ensure that the improvements are not an artefact of Synopsys Design Compiler, we ran the benchmarks through the open-source framework ABC [9] using the optimization script "resyn2rs" and obtained similar results.

However, the performance improvements are not readily predictable — improvements as large as 23% (ex1010) are possible, performance overheads of over 100% are seen in other cases. The relationship between the fraction of DCs assigned and the impact on performance metrics is also unpredictable; the minimal and maximal performance impacts on different benchmarks occur at different DC assignment fractions. In all benchmarks, complete assignment of DCs for reliability resulted in an increase in area and power consumption. In all but three benchmarks, complete assignment resulted in an increase in circuit delay. Therefore, it is desirable to incorporate metrics by which DCs can be selectively and automatically assigned to maximize reliability while simultaneously reducing the impact on performance.

3.1 Complexity factor and DC assignment

In order to better understand overhead in reliability-driven DC assignment, we introduce and explore synthetic benchmarks with varying functionality based on the functional complexity factor.



Figure 5: Normalized min, max, and mean area, power, and delay overhead across all benchmarks (y-axis) as a function of the fraction of DCs assigned for reliability (x-axis), under delay and power optimizations

Our observations lead to the development of a more sophisticated assignment — termed the complexity-factor-based DC assignment algorithm — in the subsequent section.

Table 1 reports complexity factors and DC signal probability (the fraction of minterms in the DC-set) for MCNC benchmarks. Even for these small circuits, the DC-set can encompass the vast majority of minterms. This table also reports the 'expected' complexity factor ($\mathbb{E}[C^f]$). This is the complexity factor of the function when minterms are assigned randomly based on signal probabilities:

$$\mathbb{E}[C^{f}] = (f_{0})^{2} + (f_{1})^{2} + (f_{DC})^{2}$$

where f_0 , f_1 , and f_{DC} are the signal probabilities for an incompletely specified *n*-input function:

$$f_0 = \frac{\left| \text{off-set} \right|}{2^n}, \quad f_1 = \frac{\left| \text{on-set} \right|}{2^n}, \text{ and } f_{\text{DC}} = \frac{\left| \text{DC-set} \right|}{2^n}.$$

Only one of the benchmarks has a complexity factor below 0.5 (fout), and the average complexity factor exceeds the average expected complexity factor. This indicates that benchmarks are more SOP-friendly than random chance would dictate.

Figure 6 plots normalized area against normalized error rate for 11 input, 11 output synthetic functions generated using the method described in section 2.2. For each of five families of synthetic circuits (each with the DC-set composing 60% of the total minterms), labeled by complexity factor, 10 functions were generated. Each of these functions underwent ranking-based assignment with the fraction of DCs assigned sweeping from 0 to 1. The graphed shapes describe overheads and reliability improvements associated with partial DC assignments for each family of circuits. The location (1,1) corresponds to no reliability-driven assignments. As successively more DCs are assigned, the coordinates of the resulting point will move away from (1,1), with similar circuits following similar paths. The following trends can be observed: (i) Circuits with high complexity factors have both the greatest range of improvements in error rate and the greatest area overheads for such improvements.



Figure 6: Area versus error rate for synthetic benchmarks

(ii) Circuits with lower complexity factors can achieve improvements in reliability with much smaller area overheads. (iii) Circuits with very low complexity factors can achieve simultaneous reliability and area improvements. These results point to the conclusion that low complexity factor circuits are most amenable to reliabilitydriven DC assignment.

4. Complexity-factor-based DC assignment

Although ranking-based assignment can result in reliability improvements without high overheads, it is desirable to automate the process of DC assignment to get predictable improvements in error rate for low overhead. This motivates the complexity-factor-based DC assignment algorithm, based on the derivation of a 'localized' complexity factor to guide the decision on whether or not to assign a given DC minterm.

Localized complexity factor: The results of ranking-based DC assignment on synthetic benchmarks imply that functions with a low complexity factor are most amenable to reliability-driven DC assignment. For functions with a high complexity factor, we expect those DCs that are part of low complexity factor neighborhood to minimally impact overhead under reliability-driven assignment. We can calculate the complexity factor in the neighborhood of a DC, i.e., in the region of the function encompassing minterms a short Hamming distance away from the DC. We define the normalized local complexity factor, LC^f , of a minterm x_i for an *n*-input function *f* as:

$$LC^{f}(x_{i}) = 1/n^{2} \left\{ \left\{ (x_{j}, x_{k}) \middle| \begin{array}{c} D_{H}(x_{i}, x_{j}) = 1, \\ D_{H}(x_{j}, x_{k}) = 1, \text{ and} \\ f(x_{j}) = f(x_{k}) \end{array} \right\}$$

The local complexity factor for each minterm will not be identical to the aggregate value, and we can expect a distribution of values above and below the aggregate. Based on the observations of assignment on synthetic benchmarks, we expect those minterms with the lowest local complexity factors to yield the greatest improvements in circuit reliability with the lowest area overheads. Hence, we extend the ranking-based assignment algorithm to assign only those minterms with a local complexity factor below a threshold. Based on our simulations, threshold values in the range of 0.45–0.65 avoid the high performance overheads seen in some circuits while increasing circuit reliability. Low threshold values optimize for performance, high threshold values optimize for reliability.

Table 2 reports results for the complexity-factor-based DC assignment algorithm on several synthetic as well as published bench-

```
foreach DC minterm x_i

if (LC^f(x_i) < threshold)

if (on-neighbors(x_i) > off-neighbors(x_i)) x_i \leftarrow 1

else x_i \leftarrow 0

else Leave x_i as DC
```

Figure 7: Proposed complexity-factor-based DC assignment

marks. The synthetic circuits were created with the indicated complexity factors and DC sets comprising 68.6% of the minterms, matching the average for published benchmarks. Area and error rate improvements are reported as percentages: a negative number indicates an overhead. We compare the LC^{f} -based assignments to ranking-based assignments by keeping the fraction of DCs assigned the same in both cases. This demonstrates that the LC^{j} metric is more effective at isolating DCs that avoid performance overheads. Also reported are the area and error rates that result from assigning all DCs for reliability. Although the reliability improvements offered by the LC^{f} -based assignment are not the maximum possible values, in all cases, the LC^{f} -based assignment has a smaller implementation than complete reliability-driven, or the partial ranking-based assignments. The LC^{f} -based assignment algorithm succeeds in making selections on DCs that can be advantageously assigned, or should be left for later optimizations. This is illustrated by the very high complexity factor function random3 in Table 2, where the LC^{f} -based algorithm defers to conventional assignment to avoid performance overheads.

Table 2: Complexity-factor-based assignment results

Benchmark			LC^f -based		Ranki	ng-based	Complete			
Name	i	0	C^{f}	Area	E. R.	Area	E. R.	Area	E. R.	
bench	6	8	.540	4.4	1.2	-41.2	6.1	-90.2	18.1	
fout	6	10	.338	-11.8	5.8	-12.6	5.2	-22.7	5.6	
p3	8	14	.805	1.7	1.6	2.9	0.5	-45.1	10.4	
p1	8	18	.788	-3.2	3.3	-9.0	5.7	-46.9	16.4	
exp	8	18	.788	-0.1	1.4	-5.6	1.7	-51.9	18.0	
test4	8	30	.557	-3.4	3.9	-8.6	2.5	-83.0	18.2	
ex1010	10	10	.539	4.3	21.4	-6.4	24.2	-17.9	25.5	
exam	10	10	.802	-13.7	4.5	-13.7	4.5	-121.3	49.1	
t4	12	8	.867	0.0	0.0	0.0	0.0	-5.23	14.6	
random1	12	12	.49	32.0	16.6	0.4	19.4	10.6	23.0	
random2	12	12	.667	-5.7	9.4	-9.9	5.6	-75.0	36.5	
random3	12	12	.826	0.0	0.0	0.0	0.0	-83.1	29.6	
Improvements reported as a percentage										

Nodal decomposition: The extraction of observability- and satisfiability-based DCs within a logic circuit is a well documented procedure. These DCs, distinct from the external DC sets that previous sections of this paper have dealt with, offer another avenue for application of these techniques, especially to large circuits. Before following the procedure in Figure 7, large circuits can be decomposed into smaller nodes (via, for example, the 'renode' command in ABC). The DC sets for these nodes can be extracted and analyzed in terms of their local complexity factors, then reassigned if appropriate. Working with extracted DC sets, this method increases the rate of logical masking within the circuit. With both external and extracted internal DCs available, this method increases resilience to both input and internal errors. Additionally, nodal decomposition drastically reduces the size of the circuits given to the proposed algorithms, allowing for scaling to larger circuits.

5. Reliability estimates

To understand the effectiveness of reliability-driven DC assignment algorithms, we derive min-max estimates for circuit reliability using DC assignment. These estimates can be extracted from circuit specifications without the minterm-enumerative procedures required for exact error rate computation or reliability-driven DC assignment. The minimum (maximum) error rate of a given specification, f, can be computed as the sum of base-error and mindc-error (max-dc-error). The base-error of f arises due to errors that shift the inputs to the circuit from a minterm in the off-set to a minterm in the on-set or vice-versa. Thus, base-error does not depend on the DC assignment of f. The min-dc-error (max-dcerror) is the error rate when the DCs of f are assigned to minimize (maximize) the error rate of f. For single-bit input errors and all inputs having the same probability of error, the base-error is given by the number of 1-Hamming distance minterm pairs belonging to opposite phase. The min-dc-error and max-dc-error are computed by assigning a DC to the phase that matches the phase of maximum (minimum) 1-Hamming distance neighbors. The base-error, min-dc-error, and max-dc-error can be computed as

base-error = 2
$$\left| \left\{ x_i, x_j \middle| \begin{array}{l} x_i \in \text{on-set}, \\ x_j \in \text{off-set}, \text{ and} \\ D_{\text{H}}(x_i, x_j) = 1 \end{array} \right\} \right|$$

 $\text{min-dc-error} = \sum_{x_i \in \text{DC-set}} \min(\text{on-set neighbors}, \text{off-set neighbors})$

 $\mathsf{max-dc-error} = \sum_{x_i \in \mathsf{DC-set}} \max(\mathsf{on-set} \ \mathsf{neighbors}, \mathsf{off-set} \ \mathsf{neighbors})$

where x_i , x_j are minterms of f and $D_H(x_i, x_j)$ is the Hamming distance between x_i and x_j .

Estimating the upper and lower bounds on reliability for a circuit can avoid the costly enumeration in the above formulae. We propose two methods for estimation, the first based on signal probabilities and the second on extracted parameters that encapsulate functional information. Using the signal probabilities f_0 , f_1 , and $f_{\rm DC}$, the base-error can be estimated as $2nf_0f_1$ since on average, a minterm in the off-set has nf_1 neighbors in the on-set and viceversa. Additionally, we can define a random variable Y_i for each minterm x_i as $Y_i = \sum_{j=1}^n X_j$, where X_j is a random variable for each neighboring minterm, m, of x_i such that X_j is -1/1/0 if m is in the off-set/on-set/DC-set. The statistics for random variable Y_i can be estimated using the signal probabilities f_0 , f_1 , and f_{DC} of the minterms. For functions with greater than ten inputs, by the Central Limit Theorem, Y_i can be approximated as a Gaussian random variable with mean $n(f_1-f_0)$ and variance $n(f_1+f_0-(f_1-f_0)^2)$. Y_i encapsulates information about the neighbors of minterm x_i because $\frac{n+Y_i}{2}$ is the number of neighbors in the on-set and $\frac{n-Y_i}{2}$ is the number of neighbors in the off-set. Since Y_i has a Gaussian distribution, $\frac{n-Y_i}{2}$ and $\frac{n+Y_i}{2}$ also have Gaussian distributions. Thus, the min-dc-error and max-dc-error can be estimated using [19] as

min-dc-error =
$$\mathbb{E}[\min((n - Y_i)/2, (n + Y_i)/2)]$$

max-dc-error = $\mathbb{E}[\max((n - Y_i)/2, (n + Y_i)/2)]$

These estimates assume that signal probabilities are accurate predictors of the phases of neighboring minterms. However, this is not necessarily the case: function specifications often contain larger (or smaller) implicants than signal probabilities alone can indicate. The result is that the probability of neighboring minterms sharing phase can be significantly skewed from the raw signal probability. This can be seen by comparing the expected and actual complexity factors for published benchmarks in Table 1. In order to incorporate



Figure 8: Example of border counts

structural information that can account for this, we have developed a second estimation for the error rate. It is based on the total numbers of borders in the specification, where we define a border as a pair of 1-Hamming distance minterms of different phases:

$$b_{0} = \left| \left\{ (x_{i}, x_{j}) \middle| x_{i} \in \text{off-set}, x_{j} \notin \text{off-set}, D_{H}(x_{i}, x_{j}) = 1 \right\} \right|$$

$$b_{1} = \left| \left\{ (x_{i}, x_{j}) \middle| x_{i} \in \text{on-set}, x_{j} \notin \text{on-set}, D_{H}(x_{i}, x_{j}) = 1 \right\} \right|$$

$$b_{DC} = \left| \left\{ (x_{i}, x_{j}) \middle| x_{i} \in \text{DC-set}, x_{j} \notin \text{DC-set}, D_{H}(x_{i}, x_{j}) = 1 \right\} \right|$$

Figure 8 displays the Karnaugh maps for two functions with identical signal probabilities but different border counts and consequently, different bounds on circuit reliability. Comparing these examples, we see that as the DC border count b_{DC} increases, the range of error rates possible in the circuit increases, and that as the number of 0 borders and 1 borders increases, base-error increases. Note that this information is not available when only signal probabilities are known. From the border counts, the base error rate is estimated as

base-error
$$= \frac{b_1}{2^n} \cdot \frac{f_0}{f_0 + f_{\rm DC}} + \frac{b_0}{2^n} \cdot \frac{f_1}{f_1 + f_{\rm DC}}$$
 (1)

Additionally, the expected number of borders, N_b , for each DC minterm is $b_{\rm DC}/(f_{\rm DC} \cdot 2^n)$. Hence, the expected number of on-set neighbors, $N_{\rm on}$, that each DC minterm has is $N_b b_1/(b_0+b_1)$. Here, it is possible to model the distribution of the number of on-set (offset) neighbors as a binomial distribution with $p = b_1/(b_0 + b_1)$ ($p = b_0/(b_0 + b_1)$) and $n = N_b$.

However, since N_b is the average, using a binomial distribution will disregard the range of possible values. Hence, we choose to work with the Poisson distribution, using the expected number of on-set neighbors as the λ parameter. The min-dc-error and maxdc-error terms required for min-max error estimation can be expressed as

$$\text{min-dc-error} = \sum_{i=0}^{\lfloor N_b/2 \rfloor} iP(i, N_{\text{on}}) + \sum_{i= \lfloor N_b/2 \rfloor + 1}^{N_b} (N_b - i)P(i, N_{\text{on}})$$

$$\text{max-dc-error} = \sum_{i=0}^{\lfloor N_b/2 \rfloor} (N_b - i) P(i, N_{\text{on}}) + \sum_{i=\lfloor N_b/2 \rfloor + 1}^{N_b} i P(i, N_{\text{on}})$$

where $P(k, \lambda) = \lambda^k \exp^{-\lambda} / k!$ is the Poisson distribution.

Table 3 gives the signal-probability-based and border-based estimates, as well as exact bounds for the circuit error rates. The signal probability-based estimations consistently overshoot the exact error rates, while the border-based estimations consistently contain the exact bounds.

6. Conclusions

In this paper, we showed that reliability-driven DC assignment can enhance circuit robustness through error derating. Additionally, we showed that selective reliability-driven DC assignment can

Table 3: Min-max reliability estimates

Benchmark		Error rate						Conv. Assgn.		LC^{f} -based	
Name	Gates	Ex	Exact Signal-based		Border-based		Rate	% Diff.	Rate	% Diff.	
bench	49	.117	.315	.316	.454	.041	.507	.143	22.2	.143	22.2
fout	196	.608	.850	.324	.430	.210	.514	.644	5.9	.608	0.0
p3	131	.043	.133	.351	.473	.010	.340	.048	11.6	.047	9.3
p1	195	.051	.150	.335	.467	.012	.339	.061	19.6	.059	15.7
exp	193	.050	.149	.336	.468	.012	.339	.061	22.0	.059	18.0
test4	872	.108	.276	.331	.453	.065	.404	.132	22.2	.134	24.1
ex1010	2286	.073	.139	.174	.222	.037	.179	.098	34.3	.091	24.7
exam	432	.027	.110	.398	.486	.007	.258	.053	96.3	.053	96.3
t4	76	.041	.140	.140	.361	.012	.148	.048	17.1	.048	17.1
random1	11871	.186	.294	.347	.436	.101	.281	.41	29.5	.201	8.1
random2	5055	.093	.222	.347	.436	.054	.211	.147	58.1	.133	43.0
random3	282	.078	.142	.347	.436	.021	.231	.111	42.3	.111	42.3
Average –				.149	31.8	.141	26.7				

enhance robustness and avoid the high overheads associated with complete reliability-driven DC assignment. Two algorithms — ranking-based and complexity-factor-based — for selective reliabilitydriven DC assignment based on Hamming distance and complexity factor metrics were proposed and evaluated. Finally, min-max bounds for reliability estimation of incompletely specified functions based on Hamming distance metrics were also described.

References

- S. Borkar, "Designing reliable systems from unreliable components: The challenges of transistor variability and degradation," *IEEE Micro*, vol. 25, pp. 10–16, 2005.
- [2] A. W. Strong et al., Reliability wearout mechanisms in advanced CMOS technologies. Wiley, 2009.
- [3] D. K. Pradhan, ed., Fault-tolerant computing: Theory and techniques, vol. 1. Prentice-Hall, NJ, USA, 1986.
- [4] M. Gössel and S. Graf, Error detection circuits. McGraw-Hill Book Company, London, UK, 1993.
- [5] D. P. Siewiorek and R. S. Swarz, *Reliable computer systems: Design and evalu*ation. A. K. Peters, 3 ed., 1998.
- [6] S. Mukherjee, Architecture design for soft errors. Morgan Kaufmann, MA, 2008.
- [7] J. Cong and K. Minkovich, "LUT-based FPGA technology mapping for reliability," in Proc. Design Automation Conference, pp. 517–522, 2010.
- [8] S. Hurst, D. Miller, and J. Muzio, Spectral Techniques in digital logic, ch. 4.4, p. 146. Academic Press Inc. (London) Ltd., 1985.
- [9] "ABC logic synthesis tool: http://www.eecs.berkeley.edu/~alanmi/abc/."
- [10] N. A. Touba and E. J. McCluskey, "Logic synthesis of multilevel circuits with concurrent error detection," *IEEE Trans. Computer-aided Design*, vol. 16, pp. 783–789, Jul. 1997.
- [11] S. Almukhaizim et al., "Seamless integration of SER in rewiring-based design space exploration," in Proc. Intl. Test Conference, pp. 1–9, 2006.
- [12] S. Krishnaswamy et al., "Enhancing design robustness with reliability-aware resynthesis and logic simulation," in Proc. Intl. Conference Computer-aided Design, pp. 149–155, 2007.
- [13] M. Choudhury and K. Mohanram, "Approximate logic circuits for low overhead, non-intrusive concurrent error detection," in *Proc. Design Automation and Test* in *Europe*, pp. 903–908, 2008.
- [14] K.-C. Wu *et al.*, "Soft error rate reduction using redundancy addition and removal," pp. 559–564, 2008.
- [15] N. Alves et al., "Detecting errors using multi-cycle invariance information," in Proc. Design Automation and Test in Europe, pp. 791–796, 2009.
- [16] A. Mischenko *et al.*, "Using simulation and satisfiability to compute flexibilities in Boolean networks," *IEEE Trans. Computer-aided Design*, vol. 25, pp. 743– 755, 2006.
- [17] J. P. Hansen and M. Sekine, "Synthesis by spectral translation using Boolean decision diagrams," in *Proc. Design Automation Conference*, 1996.
- [18] CUDD: Colorado University Decision Diagram package. Please visit the URL http://vlsi.colorado.edu/~fabio/CUDD/ for further details.
- [19] N. Saralees and K. Samuel, "Exact distribution of the max/min of two Gaussian random variables," *IEEE Trans. VLSI Systems*, vol. 16, no. 2, pp. 210–212, 2008.