Concurrent Error Detection for Combinational and Sequential Logic via Output Compaction

Sobeeh Almukhaizim
Electrical Engineering Dept.
Yale University
New Haven, CT 06520, USA

Petros Drineas
Computer Science Dept.
Rensselaer Polytechnic Institute
Troy, NY 12180, USA

Yiorgos Makris
Electrical Engineering Dept.
Yale University
New Haven, CT 06520, USA

Abstract

We discuss the problem of non-intrusive concurrent error detection (CED) for random logic. We analyze the optimal solution model and we point out the limitations that prevent logic synthesis from yielding a minimal cost implementation. We explain how duplication-based CED exploits decomposition to alleviate these limitations for the unrestricted error model. We then examine a compaction-based CED method, which employs a similar decomposition principle to alleviate synthesis limitations for restricted error models. We demonstrate the cost reduction achieved by the decomposed method through experimental results and we discuss the points where optimality is lost, possible remedies, and extension to finite state machines (FSMs).

1. Introduction

Concurrent test methods enable integrated circuits to verify the correctness of their results during normal operation. While this ability is highly desirable, especially in high safety applications, designing a cost-effective concurrently testable circuit is a challenging task. Quality assessment of concurrent test methods relies on several parameters, including the model of detectable faults or errors, the worst-case detection latency, and the incurred area overhead. Additionally, an important consideration is whether a concurrent test method is intrusive or non-intrusive, i.e. whether the original circuit is modified or left intact, respectively. The importance of concurrent test in only accentuated by the plethora and variety of previous research efforts in this area [1, 2, 3, 4].

Several low-cost, non-intrusive, concurrent fault detection (CFD) methods have been proposed for stuck-at faults in combinational circuits. C-BIST [3] employs input monitoring to perform concurrent self-test. While hardware overhead is very low, the method relies on an ordered appearance of all possible input vectors before a signature indicating circuit correctness can be calculated, resulting in very long detection latency. This problem is alleviated in R-CBIST [4], where the requirement for a uniquely ordered appearance of all input combinations is relaxed at the cost of a small RAM. Alternatively, latency is reduced through the comparison-based method in [5], which uses additional logic to predict the circuit responses for a complete test set.

Towards the high-cost end, several concurrent error detection (CED) zero-latency methods have been proposed for both combinational and sequential circuits [1]. Reducing the area overhead below the cost of duplication typically requires redesign of the original circuit, thus leading to intrusive methodologies. Several redesign and resynthesis methods are described in [6, 7, 8, 9], wherein parity or various unordered codes are employed to encode the states of the circuit. Limitations of [9], such as structural constraints requiring an inverter-free design, are alleviated in [10], where partitioning is employed to reduce the incurred hardware overhead. Utilization of multiple parity bits, first proposed in [11], is examined in [12] within the context of FSMs. These methods render totally self-checking circuits and guarantee zero-latency error detection; on the down side, they are intrusive and relatively expensive. Non-intrusive CED methods have also been proposed. The general algebraic model is introduced in [13]. Implementations based on Bose-Lin and Berger codes are presented in [14] and [15], respectively. Finally, parity-based CED methods for combinational circuits and FSMs are described in [1, 16]. The circuit is resynthesized to include CED based on multiple parity groups. A cost function reflecting the total cost of the modified original circuit and the parity prediction circuit guides the formation of parity groups.

In this paper, we examine a low-cost, zero-latency, non-intrusive CED method for restricted error models. The method is based on compaction of the circuit outputs, prediction of the compacted responses, and comparison. As opposed to duplication-based CED, which targets the unrestricted error model, this method achieves significant hardware cost reduction by utilizing the information available through a restricted error model. The general underlying principle of this method was first modelled in [1]. For the purpose of completeness, we review the optimal solution for this problem in Section 2. We then focus on logic synthesis limitations that prevent the optimal model from yielding a minimal cost implementation, which we demonstrate through an example in Section 3. Compaction-based CED method, which follows the paradigm of duplication-based CED and addresses these limitations through decomposition is discussed in Section 4. Experimental results in support of this method are provided in Section 5, followed by a discussion regarding optimality, possible improvements, and extension to FSMs in Section 6.

0-7695-2093-6/04 $20.00 © 2004 IEEE

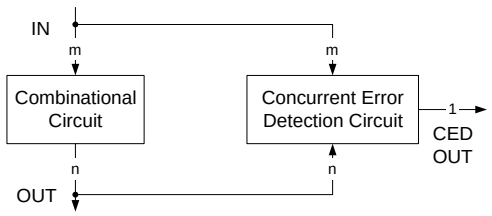

Figure 1. Non-Intrusive CED Model

2. Optimal Non-Intrusive CED

We first review the optimal model for non-intrusive CED [1]. It is important to emphasize that this discussion is based on the assumption that a restricted error model is specified. Indeed, for an unrestricted error model, wherein an error-free response may be transformed into an arbitrary erroneous response, information theory proves that any non-intrusive CED circuit will be as complex as the original circuit [17]. In this case, duplication (possibly with design diversity to avoid common-mode failures [18]) constitutes the most appropriate non-intrusive CED method.

In order to preserve generality, we assume that the error model is not defined through permanent or transient faults in the hardware, but rather in terms of the erroneous behavior that such faults induce. Thus, any fault model can be prescribed by providing for every input combination the error-free response and all erroneous responses resulting from faults in the model. Consider, for example, the combinational circuit with m inputs and n outputs shown in Fig. 1. For every input combination $a \in [0, \ldots, 2^m - 1]$, we define the error-free response of the circuit as $GM(a)$, the set of erroneous responses resulting from faults in the prescribed fault model as $BM(a)$, and the set of responses that will never occur for faults in the prescribed fault model as $DC(a)$. As depicted in Fig. 1, the non-intrusive CED circuit monitors the m inputs and n outputs and indicates errors through the 1-bit CED OUT output. The functionality of the CED circuitry may be defined for every $a \in [0, \ldots, 2^m - 1]$ as follows:

$$CED \quad OUT = \left\{ \begin{array}{rcccl} 1 & : & IN = a & \wedge & OUT = GM(a) \\ 0 & : & IN = a & \wedge & OUT \in BM(a) \\ X & : & IN = a & \wedge & OUT \in DC(a) \end{array} \right\}$$

Given this CED function definition, a synthesis tool could be employed to produce the actual CED circuit. If synthesis algorithms [19] were able to search exhaustively and generate the circuit with minimal area cost, this process would yield the optimal non-intrusive CED solution. However, in order to deal with the large search space, synthesis tools [20] employ heuristics that may lead to sub-optimal solutions. Thus, it is possible that alternative problem modelling may result in more cost-effective CED circuits.

3. Synthesis Limitations

Exact optimization methods for synthesis of circuits as multi-level networks are not considered to be practical [19]. The flexibility offered by multi-level networks in circuit implementation comes at the cost of a large search space and, consequently, great difficulty in minimizing the circuit cost. Although several heuristics exist for this purpose, their effectiveness deteriorates as the number of inputs, the number of outputs, and the number of *don't care* conditions in the circuit definition increase. An increase in the number of inputs implies a larger number of boolean sub-cubes. An increase in the number of outputs implies more opportunities for common boolean sub-cube extraction. An increase in the number of *don't care* conditions implies more flexibility in embedding a boolean function within an environment. In all three cases, the search space increases rapidly and heuristics have a hard time finding the optimal solution.

To demonstrate the impact of synthesis limitations on optimality of CED cost, we employ an example using duplication, the simplest non-intrusive CED method. More specifically, consider the 4-bit multiplier shown in Fig. 2(a). The multiplier is defined in *pla* format, synthesized using the *rugged* script of SIS [20], and mapped onto a standard library comprising 2-input gates. The cost of the multiplier is provided through the *print_map_stats* command of SIS. The hardware added for duplication-based CED includes a replica of the circuit along with an 8-bit comparator, the cost of which is also indicated in Fig. 2(a). Alternatively, as shown in Fig. 2(b), the functionality of the CED hardware can be described as a single boolean function of 16 variables, namely the 8 inputs and the 8 outputs of the multiplier. Essentially, for every input combination and corresponding error-free output the CED function is equal to 1, while for every input combination and erroneous output the CED function is equal to 0. Presumably, synthesizing the behavior of the duplicate circuit and the comparator as a monolithic entity should result in more opportunities for hardware sharing and optimization across the two modules. However, when this function is synthesized through the exact same process as above, its cost exceeds the sum of the costs of the duplicate circuit and the comparator as synthesized separately.

Clearly, this is a shortcoming of the synthesis system, which is attributed to the aforementioned reasons. Essentially, in this example, decomposed synthesis of the CED function finds a less costly solution than monolithic synthesis. For a circuit with m inputs and n outputs, monolithic synthesis searches in the space created by $m + n$ variables, while decomposed synthesis searches in a much smaller space of only m variables. Therefore, the latter is more efficient in reducing the circuit cost since it has a much smaller search space to explore. Moreover, the cost of the m-bit comparator is small, despite the fact that it is a function of $2m$ variables. The reason for this is that an m-bit comparator

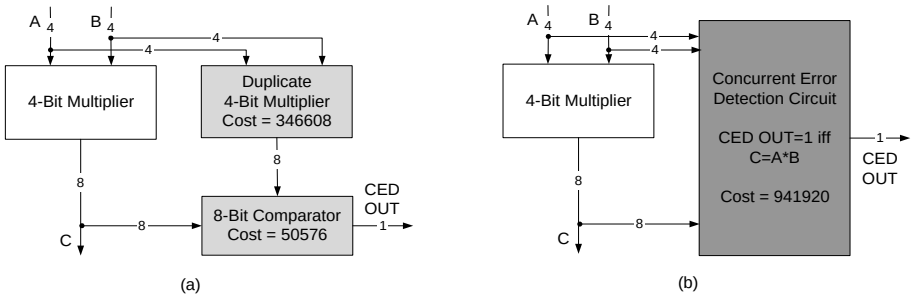

Figure 2. Decomposed vs. Monolithic Synthesis of Duplication-Based CED Circuit

is essentially a collection of m functions of 2 variables each, followed by a tree of depth $\log m$ of 2-input gates. Therefore, decomposed synthesis of duplication circuitry proves to be more cost-effective. Of course, one may not argue that this will always be the case. There is a circuit complexity threshold below which the synthesis heuristics will yield better results for the monolithic circuit. Given the NP-hard nature of the problem, however, an informed *a priori* decomposition may significantly assist the synthesis task and reduce cost.

4. Compaction-Based CED

Extending this observation to the optimal non-intrusive CED function for restricted error models discussed in Section 2, we anticipate that it will also lead to sub-optimal results. In order to alleviate this problem, we propose a method that exploits the same decomposition principle in order to reduce the cost of non-intrusive CED for restricted error models.

4.1. Methodology Overview

The proposed solution is a comparison-based, non-intrusive CED method that utilizes the information available in the restricted error model in order to reduce the area cost. More specifically, it exploits the fact that for every input a, the CED circuit needs to distinguish between the error-free response $GM(a)$ and erroneous responses in the set $BM(a)$, but not between the error-free response $GM(a)$ and erroneous responses in the set $DC(a)$, since the latter will never occur for faults in the prescribed model. As a result, the responses of the circuit may be compacted into a smaller number of bits, while preserving the information necessary to identify all errors in the restricted error model. Subsequently, it is not necessary to predict through duplication the value of all output bits and compare to the actual response of the circuit. Instead, it is sufficient to predict and compare to the compacted responses which comprise fewer bits.

The proposed scheme is depicted in Fig. 3 for a circuit with m inputs and n outputs. A compactor is added to compact the n-bit output into k bits, sufficient to distinguish between error-free and possible erroneous responses. A predictor is consequently required to predict the value of the k-bit responses for each m-bit input. Finally, a k-bit comparator is

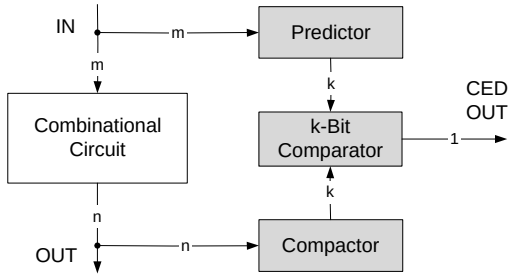

Figure 3. Compaction-Based Non-Intrusive CED

employed to indicate any discrepancy between the predicted and the actual compacted response.

In essence, this method decomposes the model of the optimal CED solution discussed in Section 2 into a compactor, a predictor, and a comparator. This can be thought of as an instance of the general scheme of *separate processing of check symbols* described in [1]. In a fashion similar to duplication, this decomposition can remedy the limitations of synthesis discussed in Section 3 and provide a low cost CED circuit. As compared to duplication, the width reduction of the predicted response is anticipated, on average, to result in a proportional cost reduction. Additionally, the size of the comparator is also reduced, leading to further cost savings. In total, and despite the additional cost incurred by the compactor, the proposed method is expected to incur less area overhead than duplication.

In terms of effectiveness, the objective of the method is to detect all errors in a restricted error model, as opposed to duplication that detects all errors in the unrestricted error model. Essentially, this is the trade-off for reducing the hardware cost of non-intrusive CED. In order to meet this objective, however, an alias-free compactor is required. Therefore, success of the proposed method relies on the ability to design such an alias-free compactor given the error-free and possible erroneous responses for each input combination.

4.2. Alias-Free Compaction

Within the context of the proposed CED method depicted in Fig. 3, a compactor is alias-free if the k-bit compacted error-free response $GM(a)$ differs from all k-bit compacted erroneous responses in $BM(a)$, $\forall a \in [0, \ldots, 2^m - 1]$. In

essence, duplication may also be viewed as an extreme case of this method, wherein $\forall a \in [0, \ldots, 2^m - 1]$, $BM(a)$ comprises all $2^n - 1$ possible erroneous responses and, therefore, k=n. Consequently, in duplication-based CED for the unrestricted error model, the compactor is eliminated and the predictor becomes a replica of the circuit. For the restricted error model, however, the set $DC(a)$, which consists of circuit responses that will never occur for faults in the prescribed model, allows alias-free compaction. The compaction function maps a group of circuit responses to the same compacted response, subject to the aforementioned constraint, and may therefore reduce the width from n to k.

In order to identify groups of compatible responses, we construct a graph $G(V, E)$, through which we will eventually derive the functionality of the compactor. The set of vertices, V, includes a vertex for every distinct error-free and erroneous response of the circuit, while the set of edges, E, includes all pairs of vertices representing error-free and erroneous responses for any circuit input. More formally:

$$V = \left\{ \bigcup_{\forall a \in [0,\ldots,2^m-1]} \{GM(a), BM(a)\} \right\} \quad and$$

$$E = \left\{ \begin{array}{c} (v_1, v_2) : v_1, v_2 \in V \quad \wedge \quad \exists a \in [0, \ldots, 2^m - 1] : \\ v_1 = GM(a) \quad \wedge \quad v_2 \in BM(a) \end{array} \right\}$$

In order to meet the constraints of an alias-free compactor, any two nodes connected by an edge in the graph need to be compacted into distinct responses. As a result, alias-free compaction reduces to the well-known graph coloring problem. More specifically, the k outputs of the compactor correspond to the bits necessary to represent the number of distinct colors of the graph. Minimization of the number of necessary colors results in minimization of k and, on average, minimization of the cost of the compactor and the predictor. We note that a similar version of this problem within the context of off-line test response compaction was also reduced to graph coloring [21]. While the problem is NP-complete, several approximation algorithms have been devised [22].

5. Experimental Results

In order to evaluate the proposed compaction-based non-intrusive CED method, we apply it on several circuits. To preserve generality, we experiment with arbitrary circuits that were generated through tables filled uniformly at random. In these experiments, the circuits have an equal number of inputs and outputs and the restricted error model comprises all errors resulting from the single stuck-at fault model. The circuits are converted in *pla* format, synthesized using the *rugged* script of SIS [20] and mapped onto a standard library comprising 2-input gates. Internally developed software employing fault simulation is used to identify the error-free and erroneous responses and, thus, to generate the conflict graph. Subsequently, the graph coloring heuristic described in [22] is used to color the nodes of the conflict graph. The functionality of the compactor and the predictor is defined through assignment of binary codes to each color, and by extension to each node in the graph. Color encoding is performed randomly. Addition of a simple comparator completes the construction of the proposed CED method.

We also constructed the 3 alternative approaches, namely decomposed synthesis of duplication-based CED, monolithic synthesis of duplication-based CED, and monolithic synthesis of the optimal CED function for restricted error models which was described in Section 2. For fairness, the same synthesis process employing the *rugged* script of SIS [20] is applied in all cases. The circuits are mapped onto a standard library comprising 2-input gates and the area cost is obtained through the *print_map_stats* command. The results are analytically presented for the components of each method and compared in the Tables of Fig. 4.

The first observation from these tables is that due to the reasons outlined in Section 3, decomposed synthesis indeed outperforms monolithic synthesis. For the unrestricted error model, decomposed synthesis of duplication-based CED costs consistently less than monolithic synthesis of duplication-based CED. Similarly, for restricted error models, the proposed decomposed CED method costs consistently less than monolithic synthesis of the optimal CED function. Therefore, the proposed decomposition of CED functionality into a compactor, a predictor, and a comparator alleviates the limitations of synthesis heuristics.

The second observation concerns the cost of the proposed method as compared to the cost of duplication. As can be seen, duplication is cheaper for the smaller circuits, while the proposed method outperforms duplication for the larger circuits. In small circuits, the conflict graph is denser, thus requiring relatively many colors. For example, in the circuit with 4 I/O, 3 bits are necessary to achieve alias-free compaction of the 4 outputs. The predictor and the comparator are now cheaper than in the case of duplication, yet not enough to compensate for the cost of the compactor. As the number of output bits increases, however, the conflict graph becomes sparser and fewer colors are needed to color it. For example, in the circuit with 7 I/O, 3 bits are still adequate to achieve alias-free compaction of the 7 outputs. In this case, the cost reduction of the predictor and the comparator surpluses the additional cost of the compactor. Interestingly, as the circuit size increases, the number of erroneous responses per error-free response appears to be growing much slower than the exponentially growing number of possible responses. As a result, conflict graphs become sparser and the number of bits necessary to color them is a diminishing proportion of the output width. Thus, as the circuit size increases, the proposed CED method is expected to provide higher savings over duplication-based CED.

	Decomposed Duplication (Unrestricted Error Model)		Proposed Compaction-Based Method (Restricted Error Models)				
Circuit	Replica	Comparator	Compactor	Predictor	Comparator	Colors	Bits
4 I/O	46864	24592	31088	39904	18096	5	3
5 I/O	100224	31088	67744	77024	18096	6	3
6 I/O	251488	37584	147552	149872	18096	8	3
7 I/O	541952	44080	220400	254272	18096	7	3
8 I/O	1332144	50576	247312	336400	11600	4	2
9 I/O	2786784	57072	558656	645424	11600	4	2
10 I/O	6265856	63568	880672	1092720	18096	5	3

	Duplication-Based CED (Unrestricted Error Model)		Compaction-Based CED (Restricted Error Models)		Comparison		
Circuit	Monolithic Duplication (MD)	Decomposed Duplication (DD)	Monolithic Optimal (MO)	Proposed Method (PM)	DD/MD	PM/MO	PM/DD
4 I/O	96976	71456	90480	89088	0.737	0.985	1.247
5 I/O	187456	131312	186992	162864	0.701	0.871	1.240
6 I/O	380994	289072	379552	315520	0.759	0.831	1.092
7 I/O	738688	586032	714096	492768	0.967	0.690	0.841
8 I/O	1879200	1382720	1904256	595312	1.013	0.313	0.431
9 I/O	5663120	2837360	3877184	1215680	0.685	0.314	0.429
10 I/O	N/T	6329424	N/T	1991488	N/T	N/T	0.314

N/T: Synthesis did not terminate within the allocated CPU time

Figure 4. Comparison of Alternative CED Methods

6. Discussion

As demonstrated above, the proposed decomposed approach outperforms monolithic synthesis and reduces the cost of non-intrusive CED for restricted error models well below the cost of duplication. Nevertheless, there are several points where optimality is lost in this method and future improvements may therefore be achieved by addressing them.

The first and most important optimality loss point concerns coloring of the graph. While the ultimate objective is the minimization of the compactor hardware, the proposed model aims at minimizing the number of colors in the graph. Even if we restrict the solution space to the minimal number of colors, a large number of alternative colorings exist. The coloring algorithm, however, does not take hardware minimization into account while selecting among them. Moreover, examining the compactor cost for each alternative coloring is infeasible. As a result, the selected coloring may lead to sub-optimal cost results. Unfortunately, this is a very hard problem requiring that hardware considerations become part of the coloring algorithm and very little is known in this area.

The second point of optimality loss concerns the assignment of binary codes to the colors of the graph. Once again, the actual color encoding impacts directly the cost of the compactor. At present, our method assigns randomly binary codes to colors, thus possibly leading to sub-optimal cost results. This problem of assigning the optimal binary codes to the colors reduces, essentially, to symbolic minimization and encoding [19] that has been studied extensively both for two-level and for multi-level logic optimization.

The third point of optimality loss concerns the decomposed synthesis of the predictor and the compactor. The aforementioned decisions on graph color assignment and color encoding affect not only the cost of the compactor but also the cost of the predictor. Optimizing these decisions for one of these two modules may adversely affect the other. One way to alleviate this problem is to compare and select among the two sequential choices, wherein decisions are tuned towards optimizing one of the two modules. To further improve the process, we are currently examining cost functions that bias color assignment and color encoding towards minimization of both the compactor and the predictor.

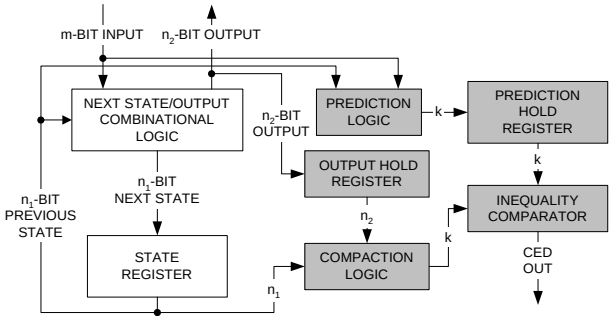

Figure 5. Extension to FSMs

Additionally, as the size of the circuit increases, exhaustively enumerating and coloring all the correct and erroneous responses to guarantee alias-free compaction becomes infeasible. In this case, a probabilistic analysis similar to [16] could be employed to provide a bound on error-masking.

On a positive note, the proposed method is readily extendible to FSMs. Controllers are typically optimized for

performance and, therefore, non-intrusive techniques are highly desirable. Consider, for example, the FSM model depicted in Fig. 5, comprising a Next State / Output combinational logic of $m + n_1$ inputs and $n_1 + n_2$ outputs, as well as a state register of n_1 bits. The proposed method may be applied directly to the Next State / Output combinational logic. However, errors caused by the State Register will not be detected. To resolve this problem, we follow the approach proposed in [12]: The n_2 output bits are held in an additional register and the compaction is performed with a latency of one clock cycle. Similarly, the prediction results are delayed by a clock cycle through a register and the comparison is performed in the next clock cycle. Errors in both the Next State / Output combinational logic and the State Register will be detected, at the cost of a constant latency of one clock cycle.

Finally, it should be noted that erroneous behavior of the added hardware will also be detected. Errors affecting the output of only the predictor or only the compactor will lead to a discrepancy between their result. Errors affecting both modules run the same danger as common mode failures in duplication, i.e. they will be detected as long as they don't mask each other, an event of relatively low occurrence probability. Similarly, as in duplication, errors in the comparator may also be detected by a totally self-checking (TSC) design.

7. Conclusion

Despite the simplicity of modelling the optimal non-intrusive CED method for restricted error models, obtaining the CED circuit through synthesis does not always yield an optimal implementation. The non-intrusive CED method proposed herein demonstrates that alternative problem modelling, which takes into account the shortcomings of synthesis, reduces significantly the cost of the CED circuit. Synthesis limitations are alleviated through a decomposed scheme, similar to duplication-based CED for the unrestricted error model. Cost reduction over duplication is achieved through alias-free compaction of circuit responses, which results in prediction and comparison of a smaller number of functions, and thus, to significantly less hardware. While limitations exist and a number of opportunities for further optimization have been identified and are currently explored, the proposed method constitutes a first step towards applicable, low-cost, non-intrusive CED for restricted error models.

References

[1] M. Gossel and S. Graf, *Error Detection Circuits*, McGraw-Hill, 1993.

[2] S. Mitra and E. J. McCluskey, "Which concurrent error detection scheme to choose?," in *International Test Conference*, 2000, pp. 985–994.

[3] K. K. Saluja, R. Sharma, and C. R. Kime, "A concurrent testing technique for digital circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 7, no. 12, pp. 1250–1260, 1988.

[4] I. Voyiatzis, A. Paschalis, D. Nikolos, and C. Halatsis, "R-CBIST: An effective RAM-based input vector monitoring concurrent BIST technique," in *International Test Conference*, 1998, pp. 918–925.

[5] R. Sharma and K. K. Saluja, "An implementation and analysis of a concurrent built-in self-test technique," in *Fault Tolerant Computing Symposium*, 1988, pp. 164–169.

[6] G. Aksenova and E. Sogomonyan, "Synthesis of built-in test circuits for automata with memory," *Automation and Remote Control*, vol. 32, no. 9, pp. 1492–1500, 1971.

[7] G. Aksenova and E. Sogomonyan, "Design of self-checking built-in check circuits for automata with memory," *Automation and Remote Control*, vol. 36, no. 7, pp. 1169–1177, 1975.

[8] S. Dhawan and R. C. De Vries, "Design of self-checking sequential machines," *IEEE Transactions on Computers*, vol. 37, no. 10, pp. 1280–1284, 1988.

[9] N. K. Jha and S.-J. Wang, "Design and synthesis of self-checking VLSI circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, no. 6, pp. 878–887, 1993.

[10] N. A. Touba and E. J. McCluskey, "Logic synthesis of multilevel circuits with concurrent error detection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16, no. 7, pp. 783–789, 1997.

[11] E. Sogomonyan, "Design of built-in self-checking monitoring circuits for combinational devices," *Automation and Remote Control*, vol. 35, no. 2, pp. 280–289, 1974.

[12] C. Zeng, N. Saxena, and E. J. McCluskey, "Finite state machine synthesis with concurrent error detection," in *International Test Conference*, 1999, pp. 672–679.

[13] V. V. Danilov, N. V. Kolesov, and B. P. Podkopaev, "An algebraic model for the hardware monitoring of automata," *Automation and Remote Control*, vol. 36, no. 6, pp. 984–991, 1975.

[14] D. Das and N. A. Touba, "Synthesis of circuits with low-cost concurrent error detection based on Bose-Lin codes," *Journal of Electronic Testing: Theory and Applications*, vol. 15, no. 2, pp. 145–155, 1999.

[15] R. A. Parekhji, G. Venkatesh, and S. D. Sherlekar, "Concurrent error detection using monitoring machines," *IEEE Design and Test of Computers*, vol. 12, no. 3, pp. 24–32, 1995.

[16] S. Tarnick, "Bounding error masking in linear output space compression schemes," in *Asian Test Symposium*, 1994, pp. 27–32.

[17] J. F. Meyer and R. J. Sundstrom, "On-line diagnosis of unrestricted faults," *IEEE Transactions on Computers*, vol. 24, no. 5, pp. 468–475, 1975.

[18] A. Avizienis and J. P. J. Kelly, "Fault tolerance by design diversity: Concepts and experiments," *IEEE Computer*, vol. 17, no. 8, pp. 67–80, 1984.

[19] G. De Micheli, *Synthesis and Optimization of Digital Circuits*, McGraw-Hill, 3rd edition, 1994.

[20] E. M. Sentovich et al., "SIS: a system for sequential circuit synthesis," ERL MEMO. No. UCB/ERL M92/41, EECS UC Berkeley CA 94720, 1992.

[21] K. Chakrabarty, B. T. Murray, and J. H. Hayes, "Optimal zero aliasing space compaction of test responses," *IEEE Transactions on Computers*, vol. 47, no. 11, pp. 1171–1187, 1998.

[22] "Network resources for coloring a graph," Available from `http://mat.gsia.cmu.edu/COLOR/color.html`.