Synthesis of Low-Cost Parity-Based Partially Self-Checking Circuits

Kartik Mohanram[†] Egor S. Sogomonyan[‡] Michael Gössel[‡] Nur A. Touba[†]

[†]Computer Engineering Research Center
University of Texas at Austin
Austin, TX 78712-1084
{kmram, touba}@ece.utexas.edu

[‡]Fault Tolerant Computing Group
University of Potsdam
14439 Potsdam, Germany
{egor, mgoessel}@cs.uni-potsdam.de

Abstract

A methodology for the synthesis of partially self-checking multilevel logic circuits with low-cost parity-based concurrent error detection (CED) is described. A subset of the inputs of the circuit is selected to realize a simple characteristic function such that CED is disabled whenever the inputs belong to the OFF-set of the characteristic function. This don't-care space in the operation of the CED circuitry is used to optimize the CED circuitry during synthesis. It is shown that this methodology is very effective at targeting faults with a high sensitization probability. Experimental results show that the proposed approach, which is of special interest in applications where a low-cost CED solution is desired, achieves a significant reduction in the error rate in logic circuits.

1 Introduction

As the complexity of modern day integrated circuits increases, testing a device for all possible faults, such as single and multiple stuck-at faults, bridging faults, and delay and crosstalk faults, poses a significant challenge for test engineers. It is widely accepted that the fault coverage for permanent faults (modeled or not) is less than 100%. In addition, studies indicate that circuits will become increasingly sensitive to temporary faults caused by crosstalk, substrate and power supply noise, charge sharing, terrestrial cosmic rays and alpha particles, etc. and that this will result in unacceptable error rates even in mainstream commercial applications. These factors have contributed to an increasing interest in methods for concurrent error detection (CED) or online detection, that can be used to monitor the behavior of the system during normal operation. Circuits with CED have the capability to detect both temporary and permanent faults and are widely used in systems where dependability and data integrity are of importance.

While efficient schemes have been developed for CED in circuits with regular structures, e.g., adders [Gorshe 96], multipliers [Pradhan 86], and PLAs [Boudjit 93], and while most memory elements use CED based on error-detecting and error-correcting codes [Chen 84], CED techniques for multilevel logic circuits have received less attention for two reasons:

(i) The very high overhead (power, area, timing, etc.) associated with conventional techniques for CED (that were developed to provide very high levels of reliability).

(ii) Lower susceptibility to temporary faults, especially at larger feature sizes (180nm and above). The reason is that when a temporary fault occurs at an internal node of a logic circuit, there are three masking factors – *logical, electrical,* and *latching-window* – that may prevent it from being latched and causing an error. In [Lidén 94], it has been shown that these three factors present a natural barrier that has prevented soft errors in logic circuits from being a major concern.

However, technology trends are causing these barriers to diminish significantly, especially for soft errors [Shivakumar 02]. The trend towards shallower logic between latches means that there is less attenuation when propagating temporary faults as well as more sensitized paths. Smaller feature sizes and lower voltage levels result in a reduction in the charge stored at a node. This allows lower energy particles to cause single event upsets capable of being latched. Lower noise margins and gate delays imply that noise transients of smaller pulse width can propagate through the circuit to the outputs. High operating frequencies mean that there are more latching windows per unit time thereby increasing the probability of a temporary error being latched. As a result of these factors, the development of CED techniques for logic circuits that meet overhead constraints is an important challenge for the future.

In this paper, we introduce a methodology for the synthesis of low-cost CED circuitry based on parity prediction for multilevel logic circuits. If the very high requirements of the totally self-checking (TSC) property [Smith 78] are relaxed, it is possible to optimize the CED circuitry while providing as high a degree of coverage as possible. The basic idea is to construct a simple Boolean function (henceforth the characteristic function) of a selected subset of the inputs to the circuit and to disable CED whenever the inputs belong to the OFF-set of this function. This constitutes a set of don't-care conditions for the operation of the CED circuitry and its synthesis is optimized with respect to the obtained don't-care set. We address two important issues with using this methodology – (1) loss in coverage, since errors caused by some faults may be undetectable (altogether, or for some input patterns) and (2) increased error detection latency, since a fault may produce errors that remain undetected for some cycles before eventual detection. We use the notion of sensitization probability of a fault to analyze both these issues and show that the proposed methodology is very effective at targeting errors caused by faults with a high sensitization probability. As the necessity for concurrent error detection in modern electronic devices continues to increase, this low cost design approach to CED with reduced area and increased latency is of special interest in the following applications:

(i) Low-cost commercial applications where coverage is desired with minimum impact to area overhead.

(ii) Applications (e.g., control circuits for electromechanical systems) where the increase in error detection latency is not critical and can hence be tolerated.

(iii) Applications where there is a sudden burst of errors spread over a short interval where the emphasis is on eventual (and not immediate) detection.

2 Low-Cost CED

The synthesis of logic circuits (e.g., control logic) with CED poses difficulties because of its inherent irregular multilevel structure. The conventional method to design logic circuits with CED is based on error-detecting codes (e.g., duplication, parity, Berger, and Bose-Lin codes). Figure 1 shows the structure of a circuit that has CED capability. The outputs of the original circuit are encoded using an error-detecting code. The check symbol generator generates check bits and the checker determines if they form a codeword. The output of the checker are the error indication bits, that are usually encoded using a two-rail code. Based upon the scheme chosen for CED, the check symbol generator can be a copy of the original circuit (duplication and compare), parity prediction logic, codeword generator (e.g., Berger or Bose-Lin codes), etc. Although duplication provides high error detection capability, the area overhead is very high (greater than 100%) and it increases with the number of outputs in the original circuit. The area overhead for CED by parity prediction ranges from 50% to 100% of the original circuit, and is higher still for the Berger code [Mitra 00]. This is because of the inherent complexity of the parity function and the inability of synthesis tools to effectively optimize large XOR-trees of Boolean functions. Thus, the basic requirements for the synthesis of TSC multilevel logic circuits results in very high overhead for most mainstream applications.

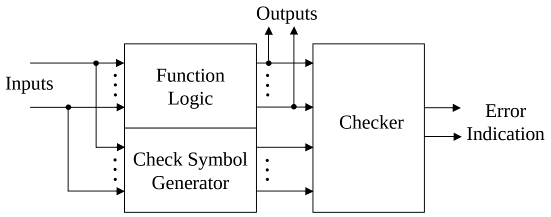


Figure 1: Block Diagram for Conventional CED

In applications where these high overheads are unacceptable, but where CED capability is desirable nevertheless, the focus shifts from designing TSC circuits to partially self-checking circuits, which satisfy the requirements of the classical definitions to the best extent possible while meeting overhead (area, power, timing) constraints. For temporary faults of a transient nature, the CED circuitry must reduce the probability that such a fault produces an undetected error. For temporary faults of an intermittent nature, the CED circuitry must reduce the probability that such a fault occurring in the circuit goes undetected the first time it produces an erroneous output. Note that there is a very high probability that a future input to the circuit causes the detection of the fault. Thus, while partially self-checking circuits cannot find use in applications where non-zero latency in the detection of errors cannot be tolerated, they are perfectly applicable where error detection with minimal latency is the primary goal.

In this paper, we propose a method for the design of low-cost parity-based CED for multilevel logic circuits. We focus on parity-based CED since the area overhead is low in comparison to other schemes based on error-detecting codes. Also, in a study of the latching probability of particle induced transients [Lidén 94], it is shown that single-bit flip model that is used in many high-level simulation procedures is reasonably accurate. This increases the relevance of parity prediction based CED schemes, since all errors with odd multiplicity (and hence single-bit errors) are guaranteed to be detected. In our approach, we select a subset of inputs to realize a characteristic function such that whenever the inputs to the circuit belong to the OFF-set of the characteristic function, the output of the parity prediction function is considered to be don't-care. The parity prediction function is optimized with respect to this set of don't-care conditions during synthesis. The subset of inputs is selected in such a way that the characteristic function is easily realized with a few logic gates and that the reduction in area is maximum.

The proposed methodology shares some similarities with a technique for the design of partially self-checking circuits described in [Wakerly 78]. The main idea described in [Wakerly 78] involves the addition of extra control signals that are used to disable the operation of the checker. Thus the checker is functional *iff* the inputs are from the appropriate input set. The differences between the approach proposed in [Wakerly 78] and that proposed here are the following:

(i) In [Wakerly 78], it is not explained where the control signals that are used to switch between the secure and insecure modes of operation are derived from. In our approach, the control signals are realized by a simple Boolean function of a carefully selected subset of the primary inputs to the circuit.

(ii) Further, the input set for which the CED circuitry is disabled is determined such that the optimization of the CED circuitry with the generated don't-care conditions results in a decrease of the area of the CED circuitry. No optimization of the CED circuitry is considered in [Wakerly 78].

(iii) All inputs and internal nodes of the logic circuit are free to change their logic values frequently. This ensures that the reduced CED implementation has a very high degree of self-testability. This is not considered in [Wakerly 78].

3 Basic Algorithm

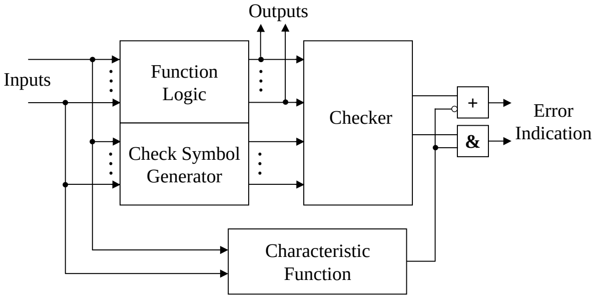


Figure 2: Proposed Approach

The basic idea of the proposed approach is illustrated in Fig. 2. A subset of the primary inputs to the circuit is chosen and used to realize the characteristic function that is used to gate the outputs (error indication bits) of the CED circuitry. Thus, whenever the inputs belong to the ON-set of the characteristic function, the results of the checking operation are available at the error detection signals of the circuit. Whenever the inputs belong to the OFF-set of the characteristic function, the outputs of the checking operation are forced to take on error-free values. Thus the inputs for which CED is disabled constitute external satisfiability don't care conditions [Hachtel 98] in the operation of the check symbol generator and the checker, i.e., the error indication signals assume error-free values independent of the output of the check symbol generator and the checker. Under the chosen fault model (single stuck-at in this paper), the synthesis of the check symbol generator (parity prediction circuitry) can be reduced to the optimal implementation of two incompletely specified Boolean functions with outputs E_1 and E_2 as shown in the shaded portion of Fig. 3. Note that it is possible to consider the characteristic function, the parity tree, and the function logic together as a single Boolean function of the primary inputs, and to optimize the same to realize the primary outputs and E_1. This is not considered in this paper.

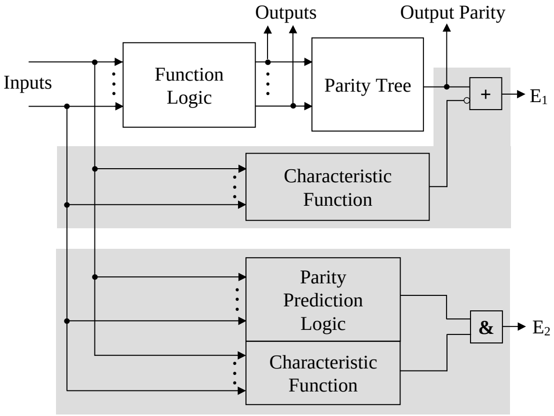


Figure 3: Partial Parity Prediction

3.1 Construction of the Disable Logic

In this section, we present the basic algorithm for identifying a suitable subset of inputs and the construction of the characteristic function. Let $\{I_1, I_2, ..., I_n\}$ be the primary inputs of the circuit. Let $\{O_1, O_2, ..., O_m\}$ be the primary outputs of the circuit. The parity prediction logic $PP(O_1, O_2, ..., O_m)$ is synthesized by optimizing the following Boolean function

$$PP(O_1, O_2, ..., O_m) \;=\; (O_1 \oplus O_2 \oplus ... \oplus O_m)$$

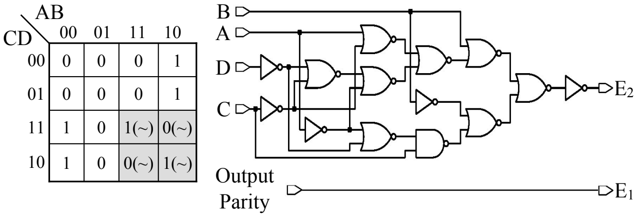
Our intuition for simplifying $PP(O_1, O_2, ..., O_m)$ to construct a partial parity prediction function $PPP(\cdot)$ of a subset of the inputs that meets cost and coverage requirements is as follows. A constant-0 or a constant-1 assignment to any of the inputs of the circuit results in several internal nodes in the circuit taking constant values. Thus, such an assignment will result in the simplification of the Boolean functions implementing the outputs. This simplification of the output functions also simplifies the parity prediction function $PP(O_1, O_2, ..., O_m)$. Since a constant-0 or a constant-1 assignment is equivalent to Boolean co-factoring, we iterate over the n inputs to this Boolean function and choose those inputs that provide the greatest simplification of the partial parity prediction function of a single input $PPP(D_i)$ given by

$$PPP(D_i) \;=\; (O_1 \oplus O_2 \oplus ... \oplus O_m) \cdot (D_i)$$

where D_i above is chosen from the set $D = \{I_1, \overline{I_1}, I_2, \overline{I_2}, ..., I_n, \overline{I_n}\}$ in succession. Once a subset of inputs S $(\subseteq D)$ that provides a large reduction in the complexity of the reduced parity prediction function is identified, we construct the partial parity prediction function of two inputs $PPP(S_i, S_j)$ given by

$$PPP(S_i, S_j) \;=\; (O_1 \oplus O_2 \oplus ... \oplus O_m) \cdot (S_i \vee S_j)$$

and resynthesize the parity prediction function for each of the identified pairs $\{S_i, S_j\}$ from the set S. Note that (1) $(S_i \vee S_j)$ constitutes a characteristic function and (2) an input may occur in both phases in S. Each of the resulting partial parity prediction functions $PPP(S_i, S_j)$ is evaluated for coverage as described in Sec. 3.2. The characteristic function that provides the largest reduction in the complexity of the $PPP(S_i, S_j)$ while meeting coverage requirements is finally chosen.



CD \ AB	00	01	11	10
00	0	0	0	1
01	0	0	0	1
11	1	0	1(~)	0(~)
10	1	0	0(~)	1(~)

Figure 4: Parity Prediction Function $PP(A, B, C, D)$ and Optimized Implementation

Construction of the characteristic function in the form described above ensures that the self-testing properties of the implementation (at the error detection signals and not the primary outputs of the circuit) is minimally affected. This is because the lines S_i and S_j can take both logic-0 and logic-1 values during normal operation without disabling the CED circuitry. We illustrate the proposed method with the aid of the example in Fig. 4, where the truth table of the parity prediction logic for some logic circuit with inputs $\{A, B, C, D\}$ is shown along with its optimized realization $PP(A, B, C, D)$. It is clear that the complexity of the function lies in the necessity to realize the minterm $(A{\cdot}B{\cdot}C{\cdot}D)$ in the truth table. When the proposed procedure is used to generate the set S, the inputs A and C in the negative phase are selected. Thus, the characteristic function is given by $(\overline{A} \vee \overline{C})$ – the don't-care space in the operation of the parity prediction logic is shown shaded in the figure. The parity prediction logic after logic optimization is given by $(A{\cdot}\overline{B} \;\vee\; \overline{B}{\cdot}C)$. When the characteristic function given by $(\overline{A} \vee \overline{C})$ is used to gate the simplified realization of the prediction logic, the final partial parity prediction function is given by

$$PPP(\overline{A}, \overline{B}) \;=\; (A{\cdot}\overline{B} \;\vee\; \overline{B}{\cdot}C){\cdot}(\overline{A} \;\vee\; \overline{C})$$

which simplifies to

$$PPP(\overline{A}, \overline{B}) \;=\; (A{\cdot}\overline{B}{\cdot}\overline{C} \;\vee\; \overline{A}{\cdot}\overline{B}{\cdot}C)$$

The optimized realization of of the partial parity prediction logic is shown in Fig. 5. It is clear that the area overhead for the proposed scheme is less than that for the original scheme. Note that in both Figs. 4 and 5, the output parity line (that drives E_1) is the output of the parity tree (driven by the outputs of the function logic).

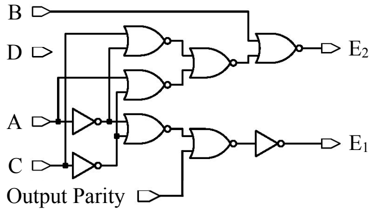


Figure 5: Partial Parity Prediction $PPP(\overline{A}, \overline{B})$

The reason for not reusing a single line to gate both error indication signals is that a single stuck-at fault on the output of the gate realizing the characteristic function line is then undetectable. This implies that single stuck-at faults on the error indication signals are undetectable. By implementing the logic that implements the characteristic Boolean function independently for the error indications signals E_1 and E_2, it is ensured that under the single fault assumption, both error indication lines can be tested by off-line test techniques.

It is also possible that the presence of faults that produce errors detectable by the proposed scheme may go undetected for a few cycles since the CED circuitry may be disabled for those combinations of inputs. This is of concern only in those applications where an intermittent fault is present in the circuit and is not applicable in the case of single event upsets produced by, say, power supply noise or terrestrial cosmic rays.

3.2 Coverage Estimation

Coverage is evaluated using two methods. The first is based upon the fault detection capability of the proposed partial parity prediction scheme in comparison to complete parity prediction. It is assumed that (1) the original circuit is intact and that no modifications have been made during synthesis to increase fault detection capability and (2) the parity prediction logic is synthesized separately from the design. Under the single fault assumption, any fault can affect either the information bits (primary outputs) or the check bits (predicted parity) but not both. A set of 32,000 pseudorandom input patterns is applied to the benchmark circuits and the fault detection capabilities of the proposed method in comparison to parity prediction is measured. For each input pattern, all the internal single stuck-at faults in *only* the function logic are injected one at a time and simulated. Note that faults in the parity prediction logic and the checker are not considered, since they are guaranteed to be detected under the single fault assumption. Considering such faults would incorrectly increase the reported coverage and make expensive CED schemes appear very effective.

If any fault is excited and propagated to the outputs of the circuit by an input pattern, it is checked whether the error at the output is detectable by parity prediction and by the proposed scheme. Thus, the coverage provided by the method is expressed as a percentage of the errors that are detectable using parity prediction:

$$\text{Coverage} = \frac{\begin{array}{c}\text{Number of errors detectable}\\ \text{by partial parity prediction}\end{array}}{\begin{array}{c}\text{Number of errors detectable}\\ \text{by parity prediction}\end{array}} \text{ x } 100\%$$

The second method for coverage estimation is based on the notion of sensitization probability for a fault. Of the three factors that affect the probability that a temporary fault produces an error, electrical and latching-window masking can be addressed by using higher drive strength gates, layout, etc. If these factors are addressed, the dominant factor is logical masking. In other words, it is important to take logical masking effects into consideration since a fault that has a sensitized path to the outputs with high probability is more likely to produce errors than one that does not. For a circuit with n inputs, the sensitization probability of a fault f with respect to parity prediction $SP_{parity}(f)$ is defined as follows:

$$SP_{parity}(f) = \frac{\begin{array}{c}\text{Number of errors detectable by}\\ \text{parity prediction produced by } f\end{array}}{2^n \text{ possible input patterns}}$$

If a fault has a high sensitization probability, then any CED method must focus on detecting errors produced by such a fault rather than on a fault that produces fewer errors at the outputs in the same period of time. Since it is not possible to determine the sensitization probability of a fault by simulating the circuit (in the presence of the fault) for all 2^n input patterns, we use 32,000 pseudorandom patterns instead. Once the sensitization probability of all the faults in the circuit is computed, we consider the difference $\Delta_{SP}(f)$ between the sensitization probability of the fault $SP_{parity}(f)$ when parity prediction is used and the sensitization probability of the fault $SP_{partial\ parity}(f)$ when the proposed scheme is used. $\Delta_{SP}(f)$ is a measure of the number of times a fault f produces an error that is not detected when the proposed scheme is used:

$$\Delta_{SP}(f) = SP_{parity}(f) - SP_{partial\ parity}(f)$$

Note that $SP_{parity}(f) \geq SP_{partial\ parity}(f)$. Hence, the greater $SP_{partial\ parity}(f)$, the smaller $\Delta_{SP}(f)$ is, and the higher the coverage that is obtained. In Sec. 4, we present experimental results that show that the partial parity prediction approach is effective at reducing $\Delta_{SP}(f)$, especially for faults with high $SP_{parity}(f)$.

4 Experimental Results

The synthesis tool used for all technology mapping and optimization in this paper is Synopsys' Design Analyzer. The technology library used is the 0.25μ library distributed by Virginia Tech [Sulistyo 02]. The tool used to implement the coverage estimation routines described in this paper is SIS [Sentovich 92].

Table 1 presents the results for CED using partial parity prediction for some combinational benchmark circuits chosen from the LGSynth91 suite [Yang 91]. Under the first major heading, we provide details about the circuits that were chosen – name, number of primary inputs, number of primary outputs, and area. Under the second major heading, we report the area of the parity prediction circuitry and the total number of errors detectable by parity prediction when the circuit is simulated with 32,000 pseudorandom patterns and every internal single stuck-at fault is injected for each of the simulated patterns. Under the third major heading, we report the area overhead and the total number of errors that are detectable using the proposed partial parity prediction scheme. The coverage (in %) of the proposed scheme with respect to parity prediction is reported in the third column under this heading. In some cases, the area overhead and coverage for more than one characteristic function that was identified is presented. The best case, emphasized in bold font, is used as the reference for Table 2. It is evident from the results that the proposed scheme reduces the area overhead while providing high coverage in all the cases.

In Table 2, we divide the interval [0, 1] into 8 equal subintervals and present the distribution of the number of faults with a sensitization probability over these subintervals. In the table, an entry of $x \rightarrow y$ under the interval $[0.125, 0.25]$ indicates that the number of faults with a sensitization probability ($= SP_{parity}$) in that interval went from x in the unprotected circuit to y ($= \Delta_{SP}$) in the circuit protected using the proposed scheme. The focus is on Δ_{SP}, since that represents errors that go undetected when the proposed approach is used. It is evident from this table that the proposed scheme targets faults with a high sensitization probability.

Table 1: Area Overhead and Coverage Results for Partial Parity Prediction

Circuit				Parity Prediction		Partial Parity Prediction		
Name	PIs	POs	Area	Area	Errors	Area	Errors	Cov. (%)
x2	10	7	1932	854	407841	**462**	**296477**	**72.7**
						504	300670	73.7
cu	14	11	2016	910	287496	**420**	**208484**	**72.5**
						434	210069	73.1
						745	214760	74.7
cm85a	11	3	1890	1792	219590	**1162**	**167994**	**76.5**
						1176	167236	76.2
ttt2	24	21	7264	8188	1579953	**5852**	**1088351**	**68.9**
						7014	1025152	64.9
						7028	1136352	71.9
x1	51	35	12569	12723	2490879	**9127**	**1698963**	**68.2**

Table 2: Sensitization Probability Distribution

Circuit	Total Faults	Interval							
		[0, .125]	[.125, .25]	[.25, .375]	[.375, .5]	[.5, .625]	[.625, .75]	[.75, .875]	[.875, 1]
x2	64	33 → 56	17 → 7	5 → 1	2 → 0	0 → 0	4 → 0	1 → 0	2 → 0
cu	52	40 → 44	3 → 8	1 → 0	0 → 0	0 → 0	0 → 0	2 → 0	6 → 0
cm85a	62	50 → 59	3 → 3	2 → 0	4 → 0	2 → 0	0 → 0	0 → 0	1 → 0
ttt2	218	82 → 193	84 → 25	22 → 0	8 → 0	1 → 0	4 → 0	16 → 0	1 → 0
x1	522	360 → 440	50 → 80	37 → 2	15 → 0	15 → 0	25 → 0	8 → 0	12 → 0

5 Conclusion

In this paper, a method for the synthesis of low-cost parity-based partially self-checking multilevel logic circuits was introduced. Experimental results show that the method provides a good cost-coverage tradeoff in comparison to parity prediction and that it targets faults with a high sensitization probability. As the necessity for CED in high-density, low-cost, high-performance computing devices increases, there is a need for such techniques.

Acknowledgements

This material is based on work completed during a visit to the University of Potsdam, Germany. It was supported in part by the University of Potsdam and in part by a grant from the Hewlett-Packard Company.

References

[Boudjit 93] M. Boudjit, *et al.*, "Automatic Generation Algorithms, Experiments and Comparisons of Self-Checking PLA Schemes using Parity Codes," *Proc. European Conf. on Design Automation,* pp. 144-150, Feb. 1993.

[Chen 84] C. L. Chen and M. Y. Hsiao, "Error-Correcting Codes for Semiconductor Memory Applications: A State-of-the-Art Review," *IBM Journal of Research and Development,* Vol. 28, No. 2, pp. 124-134, Mar. 1984.

[Gorshe 96] S. Gorshe and B. Bose, "A Self-Checking ALU Design with Efficient Codes," *Proc. IEEE Intl. VLSI Test Sym.,* pp. 157-161, 1996.

[Hachtel 98] G. D. Hachtel and F. Somenzi, *Logic Synthesis and Verification Algorithms,* Kluwer Academic Pub., 1998.

[Lidén 94] P. Lidén, *et al.*, "On Latching Probability of Particle-Induced Transients in Combinational Networks" *Proc. Intl. Sym. on Fault-Tolerant Computing,* pp. 340-349, 1994.

[Mitra 00] S. Mitra and E. J. McCluskey, "Which Concurrent Error Detection Scheme To Choose?," *Proc. IEEE Intl. Test Conf.,* pp. 985-994, 2000.

[Pradhan 86] D. K. Pradhan, *Fault Tolerant Computing – Theory and Techniques,* Vol. 1, Prentice-Hall, 1986.

[Sentovich 92] E. M. Sentovich, *et al.*, "SIS: A System for Sequential Circuit Synthesis," TRM No. UCB/ERL M92/41, Univ. of California, Berkeley, 1992.

[Shivakumar 02] P. Shivakumar, *et al.*, "Modeling the Effect of Technology Trends on the Soft Error Rate of Combinational Logic," *Proc. Intl. Conf. on Dependable Systems and Networks,* pp. 389-398, 2002.

[Smith 78] J. E. Smith and G. Metze, "Strongly Fault Secure Logic Networks," *IEEE Trans. on Computers,* Vol. C-27, pp. 491-499, Jun. 1978.

[Sulistyo 02] J. B. Sulistyo and D. S. Ha, "Developing Standard Cells for TSMC 0.25μ Technology under MOSIS DEEP Rules", Department of Electrical and Computer Engineering, Virginia Tech, TR VISC-2002-01, Jan. 2002.

[Wakerly 78] J. F. Wakerly, *Error Detecting Codes, Self-Checking Circuits and Applications,* North-Holland, 1978.

[Yang 91] S. Yang, "Logic Synthesis and Optimization Benchmarks User Guide," TR 1991-IWLS-UG-Saeyang, MCNC, Research Triangle Park, NC, Jan. 1991.