Partial Error Masking to Reduce Soft Error Failure Rate in Logic Circuits

Kartik Mohanram* and Nur A. Touba

Computer Engineering Research Center
University of Texas, Austin, TX 78712-1084
E-mail: {kmram, touba}@ece.utexas.edu

Abstract

A new methodology for designing logic circuits with partial error masking is described. The key idea is to exploit the asymmetric soft error susceptibility of nodes in a logic circuit by targeting the error masking capability towards the nodes with the highest soft error susceptibility to achieve cost-effective tradeoffs between overhead and reduction in the soft error failure rate. Such techniques can be used in cost-sensitive high volume mainstream applications to satisfy soft error failure rate requirements at minimum cost. Two reduction heuristics, cluster sharing reduction and dominant value reduction, are used to reduce the soft error failure rate significantly with a fraction of the overhead required for conventional TMR.

1. Introduction

As process technology scales below 100 nanometers, studies indicate that high-density, low-cost, high-performance integrated circuits will be increasingly susceptible to single-event upsets (SEUs) caused by high-energy neutrons (present in terrestrial cosmic radiation) and alpha particles (that originate from impurities in the packaging materials). SEUs occur when such particles strike a sensitive region in a semiconductor device, generating a dense local track of electron-hole pairs – when these electron-hole pairs are collected at a p-n junction, the current pulse of very short duration that results is termed a SEU in the signal value. A SEU may cause a bit flip in some latch or memory element thereby altering the state of the system resulting in a *soft error*. Soft errors in memories (both static and dynamic) have traditionally been a much greater concern than soft errors in logic circuits (for the same minimum feature size). However, as logic circuits move to higher operating frequencies, lower voltage levels, and smaller noise margins, it is projected that the soft error rates in logic circuits will become unacceptable even for mainstream commercial applications [Ziegler 96]. In a recent study, it has been projected that by 2011, the soft error rate in logic circuits will be comparable to that of unprotected memory elements [Shivakumar 02].

A system or component is said to *fail* if it does not correctly perform its intended function. If a soft error goes undetected, then it can result in a failure. The *failure rate* for a component or system is generally measured in units of FIT (1 failure in 10^9 hours of operation). Note that there may be other sources of failures in a system besides soft errors (e.g., permanent faults), however, this paper just focuses on the soft error failure rate (which dominates).

Two techniques that can be used to reduce the soft error failure rate are (1) error detection and retry and (2) error masking. *Error detection and retry* involves using concurrent error detection (CED) circuitry [Gössel 93], [Nicolaidis 98], [Siewiorek 98] that monitors the outputs of a circuit for the occurrence of an error. If an error is detected, the system recovers through

* Kartik Mohanram is now with the Department of Electrical and Computer Engineering, Rice University

Proceedings of the 18th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'03)
1063-6722/03 $17.00 © 2003 **IEEE**

rollback and retry thereby preventing a failure. *Error masking* involves using circuitry that masks (i.e., corrects) errors using schemes such as quadded logic [Tryon 62], interwoven logic [Pierce 65], and triple modular redundancy (TMR) [Siewiorek 98]. While error detection and retry is more commonly used, there are scenarios where error masking is advantageous. For real-time systems, it may not be possible to do retry, thus error masking is the only option. In some applications, the cost of implementing the rollback and retry functionality – state storage as well as control – may be comparable to (or exceed) that of implementing error masking. This paper focuses on error masking techniques to reduce the soft error failure rate in logic circuits.

Research on techniques for error masking has focused on mission critical applications with very high reliability requirements, where overhead cost is a secondary concern. These techniques may be overkill in the highly cost sensitive, performance oriented environment of high volume mainstream applications. Currently, beyond using parity and error correcting codes (ECC) on memories, mainstream applications incorporate little or no protection to soft errors. However, as the soft error failure rate in logic circuits increases, there is a need to develop techniques that reduce the soft error failure rate to acceptable levels at minimum cost.

In [Mohanram 03], we presented a new paradigm to synthesize CED circuitry for error detection and retry based on partial duplication of a logic circuit. It was shown that SEUs at some internal nodes in logic circuits can have orders of magnitude greater probability of being latched and causing an error than at other nodes. By focusing CED capability towards the nodes that are most susceptible to SEUs, the soft error failure rate in logic circuits can be significantly reduced at a fraction of the cost of existing techniques that try to guarantee coverage of all nodes. This paper presents a new approach based on partial error masking to reduce the soft error failure rate in logic circuits. The proposed algorithm is composed of two reduction techniques, cluster sharing reduction and dominant value reduction, and achieves a very high reduction in the estimated soft error failure rate within the specified overhead constraints. This allows cost-effective tradeoffs between overhead and soft error failure rate reduction. The proposed technique scales very well for large circuits and is highly compatible with standard synthesis flows.

2. Motivation and Previous Work

Several techniques to estimate the soft error susceptibility of nodes in a logic circuit and for the logic circuit as a whole have been presented in [Sai-Halasz 82], [Murley 96], [Hazucha 00], [Massengill 00], [Alexandrescu 02], and [Shivakumar 02]. In this paper, we utilize the model described in [Mohanram 03] to estimate the soft error failure rate of a logic circuit. The methodology can be applied fast and efficiently on a gate-level synthesized netlist of the design. After a design has been mapped to a cell library, each of the gates can be characterized with respect to their soft error susceptibility, and the interconnection of cells in the netlist can be analyzed to determine the overall soft error failure rate of the logic circuit. While radiation bombards a chip fairly uniformly in space and time, the probability that a SEU is latched varies greatly depending on which node it occurs at in the logic circuit. If a SEU occurs at an internal node of a logic circuit, there are three factors that determine whether it will be latched and result in a soft error:

1) The probability that there exists a functionally sensitized path from the node to a latch
2) The rate at which an SEU of sufficient strength to propagate to a latch occurs at a node
3) The probability that the SEU is captured in latch

The rate at which soft errors are generated at a particular latch due to SEUs at a particular node is the product of these three factors. The reader is referred to [Mohanram 03] for a

discussion on how these factors are calculated. While these three factors present a natural barrier to soft errors in logic circuits [Lidén 94], technology trends such as smaller feature sizes, lower voltage levels, higher operating frequencies, and reduced logic depth are causing these barriers to diminish significantly.

As a result of these factors, the soft error susceptibility of internal nodes (which is the contribution of the node to the overall soft error failure rate) in a logic circuit can vary by orders of magnitude. This provides an opportunity to significantly reduce the soft error failure rate at a reduced cost, since error masking can be targeted towards the nodes with high soft error susceptibility, while those with very low soft error susceptibility can essentially be ignored. This can be used to achieve a significant reduction in the soft error failure rate in a cost-effective manner, as described in Sec. 3.

3. Partial Error Masking

The partial error masking scheme proposed here is based on TMR. TMR is the simplest error masking scheme that uses three functionally equivalent copies of the logic circuit and a 2-out-of-3 majority voter. Errors are masked and hence tolerated. Figure 1 shows the structure of a circuit that has error masking capability based on TMR. The hardware overhead of conventional TMR, that targets all modeled faults in the logic circuit, exceeds 200%. The proposed method for partial error masking uses a combination of two reduction techniques, cluster sharing reduction and dominant value reduction, to reduce the overhead costs associated with TMR while minimizing the soft error failure rate. The reduction techniques are described separately in Secs. 3.1 and 3.2.

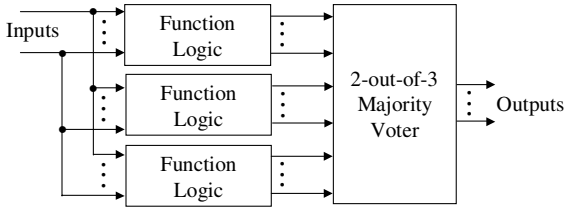

Figure 1. Block diagram for TMR-based error masking

3.1 Cluster sharing reduction

Cluster sharing reduction is based upon two observations. The first is that in the presence of the three factors mentioned in Sec. 2, the soft error susceptibility of certain nodes in the logic circuit can be orders of magnitude higher than that of the other nodes in the design. The second is that these nodes tend to be clustered together, with low observability and controllability values. Thus, the cluster sharing reduction heuristic involves selecting such clusters of nodes (in a consistent manner), so that the logic can be shared across the three copies used to realize TMR. The clusters are removed from two out of the three copies of the TMR design and they are driven by the cluster from a single copy only. The inputs to the triplicated logic of two copies of the function logic are taken from the full copy of the design. The complementary set of nodes, i.e., the nodes that are not part of any cluster and hence actually triplicated, are the ones with a high soft error failure rate. Since they will be masked by the TMR design, it is possible to achieve a significant reduction in the soft error failure rate in this manner.

If a particle strike occurs in the non-triplicated portion of the design and is of sufficient strength, it will (in the presence of a sensitized path) propagate to the outputs of all three copies and thus go undetected. However, any particle strike that occurs on a node in the triplicated logic

portion of the circuit will be masked by the 2-out-of-3 voter. By carefully selecting the clusters of nodes with low soft error susceptibility, the nodes with the highest soft error susceptibility will be in the triplicated logic portion of the circuit thereby giving a very cost-effective reduction in the soft error failure rate.

```
/* netlist – technology mapped design with soft error susceptibility data
   overhead – area overhead constraint ( overhead < area ( netlist ) )
   lowSusceptibilityQ – priority queue of gates indexed by soft error susceptibility */

while ( is_not_empty ( lowSusceptibilityQ ) || ( current_cost < overhead ) )
      node = top ( lowSusceptibilityQ ) ;
      pop ( lowSusceptibilityQ ) ;
      mark ( node ) ;
      current_cost += area ( node ) ;
      for_each_fanin ( node, fanin )
            if_not_marked ( fanin ) insert ( low_susceptibilityQ, fanin, soft_error_susceptibility ( fanin ) ) ;
      for_each_fanout ( node, fanout )
            if_not_marked ( fanout ) insert ( low_susceptibilityQ, fanout, soft_error_susceptibility ( fanout ) ) ;
      if ( overhead > threshold ) make_consistent ( network ) ;
```

Figure 2. Pseudo-code for cluster sharing reduction for error masking

A heuristic algorithm for cluster sharing to identify such clusters of logic gates with negligible soft error failure rates is presented in Fig. 2. The basic idea is to maintain the priority queue *lowSusceptibilityQ* (indexed by soft error susceptibility) of nodes that can be added to the current set of nodes in a consistent manner. Nodes with the lowest soft error susceptibility are at the head of *lowSusceptibilityQ*. It is possible that once a node is added to the cluster, all its fan-in nodes need to be added since they fan-out to only the selected node. This is accomplished by running a consistency routine to eliminate such orphan nodes by including them into the cluster. The consistency routine *make_consistent* is triggered whenever the current set of nodes grows by a certain predetermined threshold.

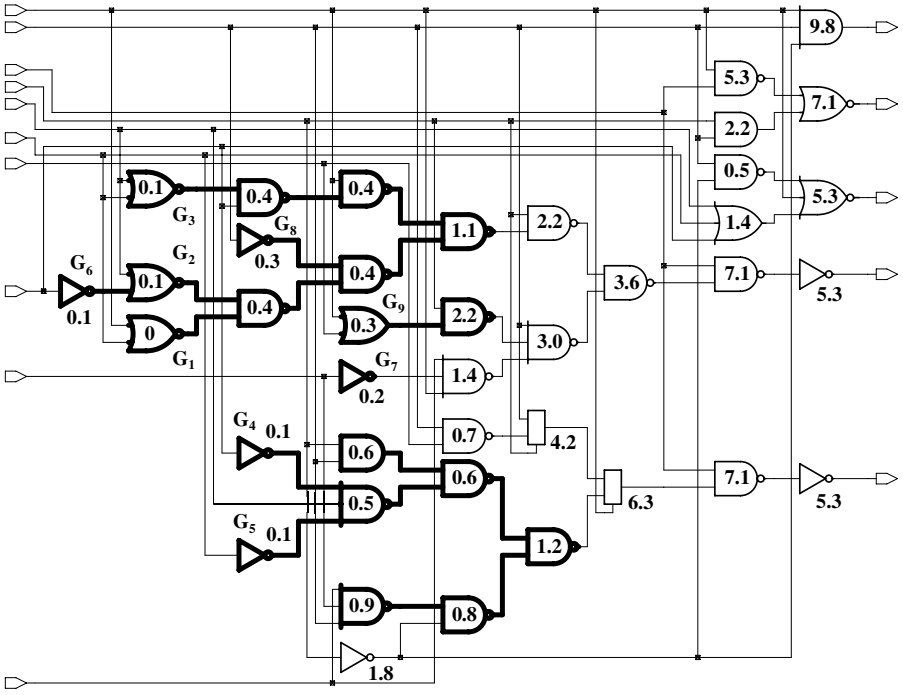

Figure 3. Screen shot with gates selected for cluster sharing

In Fig. 3, we present a screenshot of a small circuit where the gates that have been selected by the proposed cluster sharing algorithm have been highlighted. The values of the soft error susceptibility for each of the gates in the design (to 2 significant digits) is also provided. G_1 is the first gate that is popped from *lowSusceptibilityQ* and added to the cluster. The cost of the cluster is equal to the area of G_1. Its fan-in (only inputs that are ignored) and fan-out are added to *lowSusceptibilityQ*. The gates are numbered up to 9 in the order that they are added to the cluster. The overhead limit was set to 50% and all gates that are part of the final cluster are highlighted.

3.2 Dominant value reduction

Dominant value reduction differentiates between the logic 0 and logic 1 soft error susceptibility of a primary output. It exploits the fact that the logic 0 and logic 1 soft error susceptibility of certain primary outputs is highly skewed, i.e., the soft error failure rate at an output when it is at logic 0 (logic 1) is close to an order of magnitude higher that when it is at logic 1 (logic 0). The idea is to identify such outputs and replace triplication by duplication in such instances. The 2-out-of-3 majority voter is replaced by an AND (OR) logic gate.

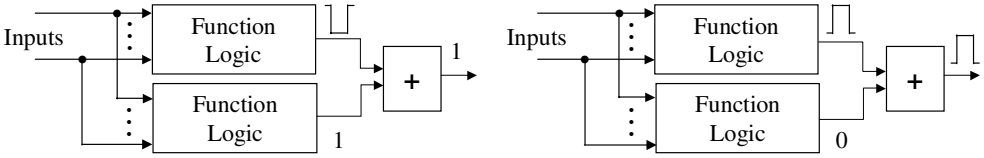

Figure 4. Example for dominant value reduction for error masking

```
Algorithm dominant_value_reduction ( netlist, max_failure_rate )
    /* netlist – technology mapped design with soft error susceptibility data
       max_failure_rate – maximum acceptable failure rate */
    list skewed_list = sort ( primary_output_list by susceptibility skew ) ;
    for_each_primary_output ( skewed_list, primary_output )
        failure_rate = evaluate_failure_rate ( netlist, dominant_value ( primary_output ) ) ;
        if ( failure_rate > max_failure_rate ) break ;
```

Figure 5. Pseudo-code for heuristic algorithm for dominant value reduction

Dominant value reduction works as follows. The soft error failure rate at the outputs of the circuit when each of the primary outputs has a logic 0 or logic 1 value are computed. For example, let primary output O_i have a logic 1 failure rate that is an order of magnitude higher that its logic 0 failure rate. Consider the approach to error masking where just two copies of the logic circuit are used and the 2-out-of-3 majority voter is replaced by an OR gate as shown in Fig. 4. Any particle strike that causes a $1 \rightarrow 0 \rightarrow 1$ transient to appear at the output O_i is guaranteed to be masked. However, a $0 \rightarrow 1 \rightarrow 0$ transient will not be masked. Thus, while the logic 1 failure rate of the primary output O_i is reduced to 0, the logic 0 failure rate is not. However, since the logic 1 failure rate is an order of magnitude higher than the logic 0 failure rate, the reduction in failure rate is:

$$\left(\frac{\text{Original Failure Rate} - \left(2 \cdot \text{Logic 0 Failure Rate}\right)}{\text{Original Failure Rate}}\right) \times 100\,\% = \left(\frac{11 - \left(2 \cdot 1\right)}{11}\right) \times 100\,\% = 81\%$$

Note that the original failure rate is the sum of the logic 0 and logic 1 failure rates for a single copy of the circuit. The logic 0 failure rate is multiplied by 2 since particle strikes to either of the copies of the logic contribute to the failure rate of the partially error masked implementation. The pseudo-code for this heuristic algorithm is presented in Fig. 5 above.

3.3 Partial error masking

The partial error masking scheme proposed in this paper utilizes a combination of the two reduction procedures that are described above to achieve a reduction in the soft error failure rate. It starts with a TMR realization for error masking that is first reduced using cluster sharing. The soft error failure rate of the resulting implementation is then estimated along with the area overhead. Note that the need to run the consistency routine does not a give precise control over the area overhead. To further reduce the area overhead, dominant value masking, where all primary outputs with a skew ratio above a specified threshold are selected is run. Since the soft error failure rate after dominant value masking is performed increases, more than one pass may be necessary before the soft error failure rate and area overhead become acceptable.

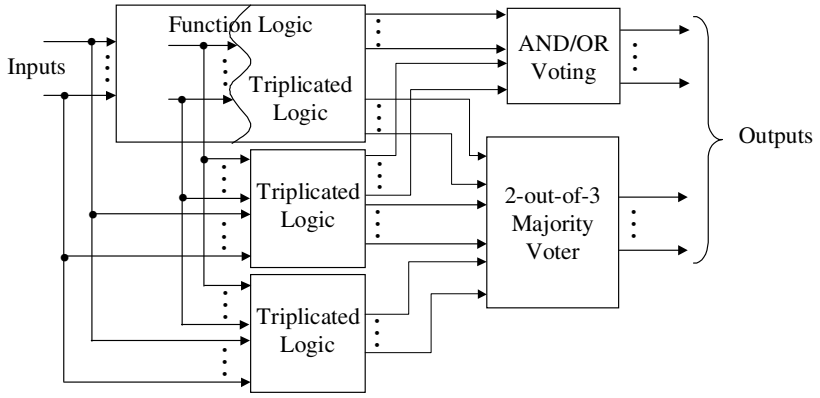

Figure 6. Partial error masking

4. Experimental Results

The synthesis tool used for all technology mapping and optimization in this paper was Synopsys' Design Analyzer. The technology library used is the 0.25 micron library distributed by Virginia Tech [Sulistyo 02]. The combinational benchmark circuits were chosen from the LGSynth91 suite [Yang 91]. The framework implementing the failure rate estimation methodology described in [Mohanram 03] was implemented in C++.

Table 1 shows the reductions in the failure rate that we achieved using the proposed cluster sharing reduction scheme by itself for partial error masking. Under the first major heading, we provide details about the circuits that were chosen – name, number of primary inputs, and number of primary outputs. Under the next two pairs of columns, we present the reduction in failure rate that was observed along with the area overhead that was necessary to achieve this. The failure rate reduction percentage was computed as:

$$\left(\frac{\text{Original Failure Rate} - \text{Reduced Failure Rate}}{\text{Original Failure Rate}}\right)\times 100\,\%$$

It is clear from the results that over an order of magnitude reduction in failure rate can be achieved with 128% overhead on average (compared with conventional TMR which requires 200% overhead). Note that with not much more overhead than what is needed for duplicate-and-compare, the proposed method corrects the errors on-the-fly thereby requiring no rollback and retry mechanism as is needed for error detection and retry schemes. Thus, it is advantageous in terms of performance (and can be used for real-time systems), moreover, in some situations the overall area overhead will be less since the rollback and retry circuitry can require significant overhead itself.

Table 1. Soft error failure rate reduction using cluster sharing reduction

Circuit			Failure Rate Reduction	Area Overhead	Failure Rate Reduction	Area Overhead
Name	No. PIs	No. POs				
C2670	233	140	95.6	129.8	86.3	103.8
C3540	50	22	90.4	127.2	83.1	103.7
C5315	178	123	85.9	126.6	82.6	104.0
C7552	207	108	95.0	128.8	89.8	103.7
x1	51	35	95.4	128.4	93.7	103.8
c880	60	26	89.6	128	83.5	99.8
b9	41	21	96.3	128.8	91.1	101.6
Average			92.6	128.2	87.2	102.9

Figure 7 shows the reductions in the soft error failure rate achieved using dominant value reduction by itself. On the X-axis, the number of skewed outputs that were chosen increases from 0 to 10. When no output is chosen, the overhead is 200%, since two full copies of the logic circuit are necessary. (Note that we ignore the overhead of the voter throughout.) As more outputs are selected for dominant value masking, one of the three copies of the logic circuit no longer requires the logic that belongs exclusively to the transitive fan-in cone of the dominant value masked primary outputs. Thus, the overhead ranges from 200% (no primary outputs chosen) to 100% (all outputs chosen). Note that the effectiveness of dominant value reduction varies considerably from circuit to circuit. This is because it depends on the extent of the signal probability skew and the amount of logic sharing between the relevant output cones.

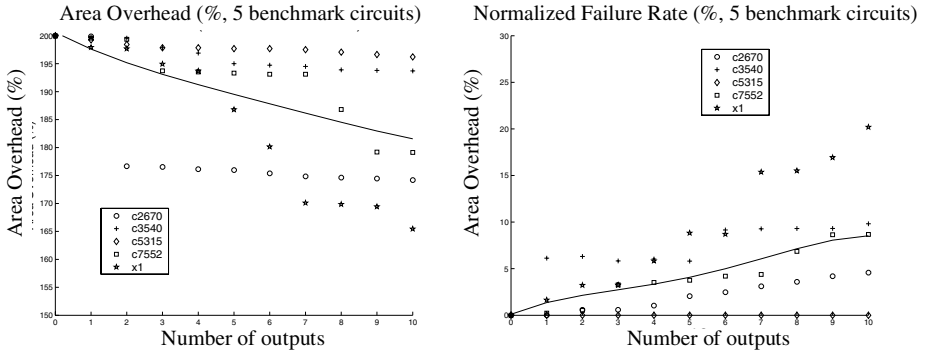

Figure 7. Dominant value reduction – outputs vs. overhead and soft error failure rate reduction

Table 2 shows the reductions in the soft error failure rate and overhead achieved using the partial error masking scheme described in Sec. 3.3. Under the first major heading, we provide details about the circuits – name, number of primary inputs, and number of primary outputs. Under the next two pairs of columns show the reduction in soft error failure rate that was observed along with the overhead that was necessary to achieve this. In both cases, we ran cluster sharing reduction to reduce the circuit to the soft error failure rate and overhead described in Table 1. This was followed by dominant value reduction, where 10 outputs with the largest skew were selected. In our experiments, we observed the optimum number of outputs that need to be selected for dominant value reduction varies from one logic circuit to the other. For example, *C2670* showed a 86.3% reduction in the soft error failure rate with 103.8% overhead when only cluster sharing reduction is applied. However, when partial error masking is performed, we see that a 91.0% reduction in soft error failure rate can be achieved with 104% overhead. In this case, the combination of cluster sharing and dominant value reductions is clearly advantageous. A similar observation holds for the benchmark *C5315*. The partial error

masking algorithm uses an iterative process (of cluster sharing reduction and dominant value reduction) to meet soft error failure rate and overhead requirements.

Table 2. Soft error failure rate reduction using partial error masking

Circuit			Failure Rate Reduction	Area Overhead	Failure Rate Reduction	Area Overhead
Name	No. PIs	No. POs				
C2670	233	140	91.0	104.0	81.7	78.0
C3540	50	22	81.6	120.9	74.7	97.4
C5315	178	123	85.9	122.8	82.6	100.2
C7552	207	108	86.6	108.0	81.2	82.9
x1	51	35	77.4	94.0	75.7	69.4
c880	60	26	85.0	124	79.0	95.8
b9	41	21	71.3	77.8	66.1	50.5
Average			82.7	107.4	77.3	82.1

5. Conclusions

In the future, as the soft error failure rate of logic circuits becomes unacceptably high even for mainstream applications, it will become necessary to incorporate error masking features into logic circuits. In some applications, error masking is advantageous to error detection and retry. For such applications, this paper described a partial error masking scheme to realize cost-effective fault tolerance by exploiting the asymmetric soft error susceptibilities of nodes.

References

[Alexandrescu 02] D. Alexandrescu, *et al.*, "New Methods for Evaluating the Impact of Single Event Transients in VDSM ICs," *Proc. Defect and Fault Tolerance Symposium,* pp. 99-107, 2002.

[Gössel 93] Gössel, M. and Graf, S., *Error Detection Circuits*, McGraw-Hill Book Co., London, 1993.

[Hazucha 00] P. Hazucha and C. Svensson, "Impact of CMOS Technology Scaling on the Atmospheric Neutron Soft Error Rate," *IEEE Trans. Nuclear Science*, Vol. 47, Iss. 6, pp. 2586-2594, Dec. 2000.

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

[Massengill 00] L. W. Massengill, *et al.*, "Analysis of Single-Event Effects in Combinational Logic Simulation of the AM2901 Bitslice Processor," *IEEE Trans. Nuclear Science*, Vol. 47, Iss. 6, pp. 2609-2615, Dec. 2000

[Mohanram 03] K. Mohanram and N. A. Touba, "Cost-Effective Approach for Reducing Soft Error Failure Rate in Logic Circuits,"' (to appear in) *IEEE International Test Conference,* 2003.

[Murley 96] P. C. Murley and G. R. Srinivasan, "Soft-Error Monte Carlo Modeling Program, SEMM," *IBM Journal of Research and Development,* Vol. 40, No. 1, pp. 109-118, 1996.

[Nicolaidis 98] M. Nicolaidis and Y. Zorian, "Online Testing for VLSI – A Compendium of Approaches," *Jnl. Electronic Testing: Theory and Applications,* Vol. 12, Nos. 1/2, pp. 7-20, Feb.-Apr. 1998.

[Pierce 65] W. H. Pierce, *Failure-Tolerant Computer Design,* Academic Press, New York, 1965.

[Sai-Halasz 82] G. A. Sai-Halasz, *et al.*, "Alpha-Particle-Induced Soft Error Rate in VLSI Circuits," *IEEE Jnl. Solid-State Circuits,* Vol. 17, No. 2, pp. 355-361, Apr. 1982.

[Siewiorek 98] D. P. Siewiorek and R. S. Swarz, *Reliable Computer Systems: Design and Evaluation* (3rd edition), A. K. Peters, 1998.

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

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

[Tryon 62] J. G. Tryon, "Quadded Logic," *Redundancy Techniques for Computing Systems,* pp. 205-228, Spartan, 1962.

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

[Ziegler 96] J. F. Ziegler, *et al.*, "IBM Experiments in Soft Fails in Computer Electronics (1978-1994)," *IBM Jnl. Research and Development*, Vol. 40, pp. 3-18, 1996.