Input Ordering in Concurrent Checkers to Reduce Power Consumption

Kartik Mohanram and Nur A. Touba

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

Abstract

A novel approach for reducing power consumption in checkers used for concurrent error detection is presented. Spatial correlations between the outputs of the circuit that drives the primary inputs of the checker are analyzed to order them such that switching activity (and hence power consumption) in the checker is minimized. The reduction in power consumption comes at no additional impact to area or performance and does not require any alteration to the design flow. Since the number of possible input orders increases exponentially in the number of inputs to the checker, the computational costs of determining the optimum order can be very expensive. We present a very effective technique to build a reduced cost function to solve the optimization problem to find a near optimal order.

1. Introduction

As process technology scales below 100 nanometers, high-density, low-cost, high-performance integrated circuits, characterized by high operating frequencies, low voltage levels, and small noise margins will be increasingly susceptible to temporary faults. Temporary faults include those caused by crosstalk, substrate and power supply noise, charge sharing, etc., and pose a significant challenge to ensuring signal integrity even in present-day deep sub-micron process technologies. In addition, current studies indicate that circuits will become increasingly sensitive to temporary faults caused by terrestrial cosmic rays and alpha particles (that originate from impurities in the packaging materials), and that this will result in unacceptable soft error rates even in mainstream commercial electronics [Ziegler 96], [Cohen 99]. Circuits with concurrent error detection (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 [Nicolaidis 98]. While software implemented error detection and correction schemes are available, they are not ideal in situations where early detection of errors is critical for preserving the state of the system and maintaining data integrity. Circuit-level techniques for CED permit early detection and containment of errors before they can propagate to other parts of the system and corrupt data. This not only reduces the complexity but also increases the effectiveness of system-level and application-level fault tolerance features.

High dependability comes at a cost however, since CED schemes impose extra overhead (area, timing, and power) on the design. While the necessity to incorporate CED schemes in at least the critical sections of a design cannot be stressed enough, there is a strong need for techniques that will make the overhead costs of incorporating them acceptable. Thus, meeting high dependability requirements with minimal overhead costs is a significant challenge facing the research community. While considerable effort has been directed towards reducing the area and delay penalties associated with CED schemes, little effort has been directed towards reducing the associated power consumption. Power, for long a concern confined to the realm

of portable systems, is today a first order factor influencing integrated circuit design at all levels of the design flow. Power issues also contribute to reliability concerns through electromigration and hot-electron degradation effects.

In this paper, we present an input ordering algorithm that can be used to reduce power consumption in checkers used for CED. Checkers used for CED are functionally symmetric with respect to their inputs, and hence the inputs to the checker can be connected in any arbitrary order. The proposed approach exploits this functional symmetry, and the spatially correlated nature of the inputs to the checker to order the inputs such that switching activity, and hence power consumption, in the checker is reduced. The main advantage of this approach is that there are no overhead costs, and no modifications are required either to the checker or the design. The only cost is the time for computing the input ordering for the checker that minimizes the power consumption. Since the number of possible input orders is exponential in the number of inputs to the checker, the optimization problem of determining the optimal input order is computationally expensive even for a small number of inputs. We present a fast heuristic method to determine an input order that is near optimal with respect to reduction in power consumption.

2. Concurrent Checkers and Input Ordering

Conventional schemes to design circuits with CED based on error-detecting codes such as parity, duplicate-and-compare, etc., employ checkers to monitor the outputs for the occurrence of an error. Figure 1 shows the structure of a circuit that has CED capability. 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., for Berger or Bose-Lin codes), etc. The check symbol generator generates check bits and the checker determines if they form a codeword. Checkers can have an adverse affect on timing (i.e., increase the clock cycle time), and hence they are usually pipelined by adding latches right before the checker. By deferring the checker operation to the next clock cycle, the performance of the design remains unaltered at the cost of some latency on when the error is detected. An error occurring in one clock cycle is detected in the next clock cycle. Generally, this is early enough to prevent data corruption in most applications.
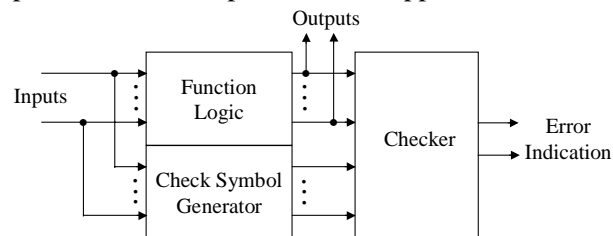

Figure 1. Block diagram for conventional CED

As indicated in Fig. 1, concurrent checkers for error detecting codes usually derive their inputs from the primary outputs of a circuit. These outputs can be connected to the inputs of the checker in any order, since most checkers have a regular structure and are functionally symmetric with respect to their inputs. In other words, the functionality of the checker is insensitive to various permutations (orderings) of the inputs.

In the presence of spatial and temporal correlations in the primary outputs of the circuit driving the checker, the input ordering presented to the checker can have a significant effect on power consumption in the checker. Correlation between vectors is of two types – spatial and temporal. Spatial correlation refers to the correlation between pairs of bits in the same input vector, while temporal correlation refers to the correlation between pairs of input vectors,

spaced one or more cycles apart. Power estimation techniques usually make the spatial independence assumption about the primary inputs to a circuit and neglect spatial correlations between them [Najm 94]. A recent study of industrial circuits [Schneider 96] evaluated the accuracy of the correlation assumptions made by several power estimation methods. Large inaccuracies in total switching activity are reported when correlation between signals is neglected. Assuming spatial independence provides a very conservative estimate of power consumption for checkers. It also precludes the possibility that the inputs to the checker can be ordered to reduce power consumption. Given their symmetry, the grouping of the inputs to the checker can affect power consumption to a considerable extent due to spatial correlations in the inputs. In this paper, we address the problem of input ordering to checkers to minimize power consumption and present an algorithm that achieves this while taking spatial input correlations into consideration.

2.1 Previous Work

Previous techniques that use reordering to reduce power consumption in CMOS gates work at the input and transistor levels. Input reordering methods permute only the inputs to the gates, while leaving the actual realization of the gate untouched. Transistor reordering methods modify the order in which series transistors are connected in a complex CMOS gate, in addition to input reordering. These methods list all possible configurations of a complex CMOS gate and evaluate them for power consumption. The number of configurations that are evaluated is usually small. There is usually a delay tradeoff involved in such techniques, since reordering can move late arriving inputs farther away from the output of the gate contributing to an increase in delay. In [Prasad 96], a multi-pass transistor reordering algorithm, with linear time complexity per pass, that converges to a solution in a small number of passes was presented.

Transistor resizing techniques resize transistors subject to delay constraints to reduce power consumption. These techniques compute the slack at each gate in the circuit and process those with a positive slack. The sizes of the transistors in such gates are reduced until the slack reaches zero or the transistors reach minimum size. In [Tan 94], an algorithm that combines both the input reordering and transistor resizing approaches is presented.

The proposed approach differs from previous reordering methods in several ways. Previous methods target general circuits and focus on input and transistor reordering at the individual gate level, whereas the proposed method targets checker circuits and focuses on input reordering at the module level. The magnitude of the problem addressed here is larger, since *n* inputs to a module imply *n!* possible permutations, however, the potential for improvement is much greater. Even after symmetry is accounted for, the number of distinct permutations renders prohibitive the costs of exhaustive enumeration and evaluation.

There has also been some work on the synthesis of checkers with low power consumption. In [Metra 96], a methodology to design tree checkers with low power-delay requirements was presented. In [Kavousianos 98], a methodology for designing Berger-code checkers with near optimal transistor count, high speed, and low power consumption was presented. The proposed approach differs from these methods, since it does *not* consider the design of the checker in isolation from the circuit that drives it.

Thus the proposed approach can be used to obtain an optimal permutation that reduces power consumption in the checker. The reordering and checker synthesis techniques described above can be used independently to obtain further reductions in power consumption, especially in the presence of structural asymmetries in the checker. The proposed approach thus complements other low power synthesis techniques, whether they target general circuits or checkers.

If the checkers are pipelined, all the inputs to the checker are ready at the same instant of time. In the absence of such pipelining, an extra dimension is added to the complexity of the problem. Even in the most balanced of designs, different paths have different delays, and the inputs to the checker will be ready at different times. This can contribute to glitches in the checker, since its inputs are not ready at the same instant, and the extra switching activity can increase power consumption. This is not addressed here.

3. Proposed Methodology

We use parity codes as an example to illustrate the key idea of our contribution. We present a small example on how spatial correlations in the inputs can affect power consumption in the checker. Consider the Boolean functions $F_1, F_2, \ldots$ and F_7 in Fig. 2. These (in some permutation) drive the inputs $x_1, x_2, \ldots$ and x_7 of the parity tree shown in Fig. 3.

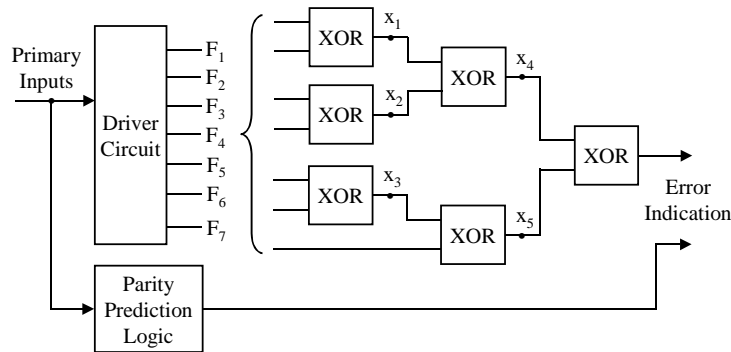Figure 2. Boolean Functions F_1, F_2, $\ldots$ and F_7

Figure 3. Example Circuit with Parity Encoded Outputs

A naïve approach to solving the problem of determining the optimum ordering of the inputs to the checker would be to exhaustively enumerate all possible permutations and to compute the exact power consumption for each of the possible solutions. The best permutation is then chosen. For the example in Fig. 3, the optimum permutation obtained by exhaustive enumeration has a power consumption of 106 units. The computational costs of this method are exorbitant even for small values of *n,* since the number of possible permutations is *n!* (and hence exponential in *n*). We propose a simulated annealing approach to solve the optimization problem of determining the best permutation. Simulated annealing that uses the Metropolis Monte Carlo algorithm and a logarithmic cooling schedule is used [Kirkpatrick 83]. We present two methods, both of which use the same simulated annealing framework, but differ in the complexity of the chosen cost function.

3.1 Exact Simulation Method

The first method, which is computationally expensive, uses the exact routine to calculate the power consumption in the checker as the cost function. This involves the use of a power estimator that simulates the checker using an output trace of the circuit driving the checker and computes the transitions at each of the internal nodes of the checker to estimate power. We

term this method the "exact simulation method". For the example in Fig. 3, the optimum permutation returned by the exact simulation method has a power consumption of 106 units. However, making a call to the power estimator for each permutation encountered during the simulated annealing routine is very expensive. The total number of calls is equal to the number of permutations encountered per iteration multiplied by the number of times the temperature is reduced during the simulated annealing routine. This can result in very high computational costs as the number of inputs to the checker increases.

3.2 Spatial Correlation Estimation Method

The second algorithm uses the same simulated annealing framework as the exact simulation method, but with a reduced cost function that results in a substantial decrease in the runtime complexity. We term this method the "spatial correlation estimation method". For the example in Fig. 3, the optimum permutation returned by the spatial correlation estimation method has a power consumption of 106 units. This method is so called because the reduced cost function is built using the values of spatial correlation that are computed for the outputs of the circuit driving the checker. The use of the reduced cost function does not involve any circuit simulation for each permutation encountered during the simulated annealing routine. The reduced cost function is built *once* by a structural analysis of the checker, and all input orders are evaluated by simple substitution of spatial correlation values into this function. This avoids the use of the power estimator that is the computational bottleneck of the exact simulation method. The pseudo-code for the spatial correlation estimation method is presented in Fig. 4.

```
Algorithm Compute_Ordering ( checker_netlist, patterns ) {

        /* patterns – derived from the output trace of the circuit that drives the checker
            checker_netlist – technology mapped version of the checker */

        correlation = compute_correlation ( patterns );     /* compute spatial correlation between all pairs of output bits */

        reduced_cost_function = build_cost_function ( checker_netlist, correlation );
        /* build reduced cost function using the checker netlist and computed values for spatial correlation */

        permutation = random ( );     /* random initial permutation */

        permutation = simulated_annealing ( reduced_cost_function, permutation );
        /* Simulated Annealing routine to solve the optimization problem */

        return permutation;
}
```

Figure 4. Pseudo-code for the spatial correlation estimation method

The reduced cost function uses the notion of the transition probability at a node, used by probabilistic techniques for power estimation, as a measure of the average switching activity at that node [Najm 94]. The transition probability $P_t(x)$ of a node x in a circuit corresponds to the average fraction of clock cycles in which the steady state value of the node differs from its initial value. Thus, reducing the transition probability at a node has the direct benefit of reducing the power consumption at that node. In addition, it is likely that this reduction in switching activity results in a reduction in the switching activity at the nodes that depend on this node. This concept can be extended to the checker as a whole, since an optimal permutation will certainly reduce switching activity at nodes close to the primary inputs of the checker. This has a cascading effect, in that fewer transitions occur at the outputs of the gates that use these nodes as inputs, and so on to the primary outputs of the checker.

The reduced cost function is built by a structural analysis of the checker which is decomposed using 2-input gates. To build the reduced cost function, we compute the exact transition probability for all those signals that depend on one or two primary inputs to the checker. This is usually possible for nodes that reach a topological depth of two or three in the checker. The transition probability at the node is weighted by the load capacitance driven by

that node. The reduced cost function for the example in Fig. 3 (assuming unit load capacitance) is shown in Fig. 5. Note that the reduced cost function does not include the transition probabilities at nodes x_4 and x_5, since they depend on more than two primary inputs.
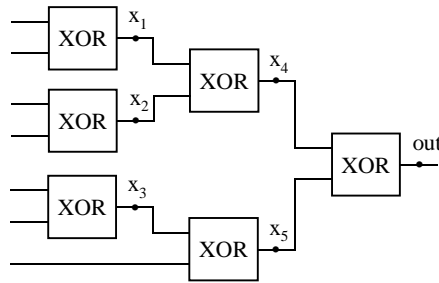
Reduced Cost Function $= P_t (x_1) + P_t (x_2) + P_t (x_3)$

Figure 5. Reduced Cost Function for the example from Fig. 3.

It is important to use spatial correlation values to compute the transition probability at nodes while building the reduced cost function. It is possible to use, under the spatial independence assumption, the signal probability at a node to estimate the transition probability at that node [Najm 94]. This is not as efficient, however, since correlations between pairs of inputs are not captured with sufficient accuracy. This is illustrated with an example in Fig. 6. Consider an XOR gate, driven by the functions F_1 and F_2 from Fig. 2. In Fig. 6, we compare the transition probability at the output of the XOR gate computed when spatial independence is assumed with the exact transition probability. Note that there is a significant difference (0.50 versus 0.22).

Transition at output if inputs switch   { 00, 11 } ⬌ { 01, 10 }

$P (\text{output} = 1)_{\text{spatial independence}} = 0.50$ $P (\text{output} = 1)_{\text{exact}} = 0.125$

$P_{\text{transition}} (\text{output})_{\text{spatial independence}} = 0.50$ $P_{\text{transition}} (\text{output})_{\text{exact}} = 0.22$
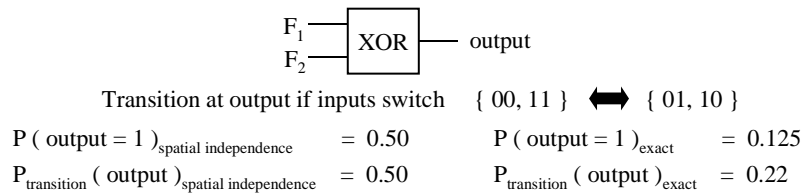
Figure 6. Inaccuracies when spatial independence is assumed

If the transition probabilities at the nodes are not accurate, the reduced cost function is not accurate. Such discrepancies can seriously affect the direction taken by the spatial correlation estimation method. For the example in Fig. 3, a solution that is not close to the optimum is obtained when spatial correlations are neglected, i.e., when spatial independence is assumed. The final permutation returned by the spatial correlation estimation method using the reduced cost function built under the spatial independence assumption has a power consumption of 119 units. This is not only considerably off the global optimum of 106 units, but also very close to the maximum power consumption that was observed (126 units).

Correlations between all pairs of outputs of the circuit driving the checker are estimated by doing random vector simulation of the circuit driving the checker. Random vector simulation is replaced by actual application vector simulation when they are available, since this best captures the correlations (both spatial and temporal) between signals. For every pair of outputs (F_i, F_j) of the driver circuit, there are four combinations of values that can occur depending on the inputs to the circuit – *00, 01, 10,* and *11* – and hence four values of spatial correlation (that sum to 1) to be computed. Once the correlation values are known, the exact transition probability at a node (that depends on two primary inputs) can be directly computed.

The main benefit of using the reduced cost function is that many more permutations can be explored with minimal trade-off in the accuracy and quality of the final permutation that is obtained. In Sec. 5 we present experimental results that indicate that the power consumption of the final permutation obtained is within 10% of the optimum permutation computed using the exact simulation method for all the test cases. We also present the runtimes that clearly show that the spatial correlation estimation method is at least an order of magnitude faster than the exact simulation method.

4. Implementation and Experimental Results

The synthesis tool used for all technology mapping and power estimation in this paper is SIS [Sentovich 92]. Some combinational benchmark circuits were chosen from the LGSynth91 suite [Yang 91]. 100,000 random vectors were used to obtain an output trace from each of the circuits. This trace was used to compute the spatial correlation between the outputs, as well as the power consumption for each of the permutations that were evaluated (for the exact simulation method). Separate runs using the *minimal.genlib* and *mcnc.genlib* technology libraries were performed, since the optimal permutation obtained, as well as the reduction in power consumption achieved, varies according to the library used for technology mapping.

Table 1 presents the results for parity checkers for some combinational benchmark circuits chosen from the LGSynth91 suite, mapped using the *minimal.genlib* technology library. 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 second major heading, we report the average power consumption over 100 random input orderings that were used to drive the checker. Under the third major heading, we provide the power consumption for the optimal permutation obtained using the exact simulation method, as well as the runtime. Under the fourth major heading, we provide the power consumption for the optimal permutation obtained using the spatial correlation estimation method, as well as the runtime. It is evident from the results that reordering the inputs results in significant power consumption reduction when parity checkers are used. In addition, the spatial correlation estimation method provides a near optimal solution at a fraction of the computational cost of the exact simulation method.

Table 1. Power Consumption Reduction Results for Parity Checkers for some Combinational Circuits when the *minimal.genlib* library is used

Circuit			Average Power (100 random orderings)	Exact Simulation Method		Spatial Correlation Estimation Method	
Name	Num PIs	Num Pos		Power	Runtime (seconds)	Power	Runtime (seconds)
x2	10	7	80	72	1066	73	16
cu	14	11	76	60	2368	63	31
sct	19	15	289	262	3286	274	53
b9	41	21	432	394	6119	425	97
seq	41	35	455	347	13400	347	246
x1	51	35	830	725	13410	746	249
vda	17	39	867	710	14672	728	303
k2	45	45	572	*	*	444	396
i5	133	66	2230	*	*	2118	872
i6	49	67	2334	*	*	2109	886

Table 2 presents the results for parity checkers mapped using the *mcnc.genblib* technology library. The results are similar to those obtained when the *minimal.genlib* technology library is used – there is a reduction in power consumption, and the spatial correlation estimation method returns a solution that is close to that returned by the exact simulation method at a fraction of the computational cost.

Table 2. Power Consumption Reduction Results for Parity Checkers for some Combinational Circuits when the *mcnc.genlib* technology library is used

Circuit			Average Power (100 random orderings)	Exact Simulation Method		Spatial Correlation Estimation Method	
Name	Num PIs	Num Pos		Power	Runtime (seconds)	Power	Runtime (seconds)
x2	10	7	44	35	188	36	16
cu	14	11	43	31	374	34	31
sct	19	15	131	111	468	118	53
b9	41	21	207	140	769	152	97
seq	41	35	212	113	1300	122	246
x1	51	35	374	315	1298	349	249
vda	17	39	381	275	1394	298	303
k2	45	45	281	188	1579	198	396
i5	133	66	786	*	*	764	872
i6	49	67	857	*	*	776	886

Table 3 presents the results for Berger-code checkers for some combinational benchmark circuits chosen from the LGSynth91 suite, mapped using the *minimal.genlib* technology library. Berger-codes are a class of systematic unidirectional error detecting codes [Pradhan 86]. Berger code checkers are of two types in practice, based on parallel ones counters and sorting networks [Piestrak 01]. We focus on ordering the inputs to the parallel ones counter to reduce power consumption, since they have been shown to be smaller, faster, and lower on power consumption as the number of inputs increases [Piestrak 01]. The results are similar to those obtained for parity checkers – there is a reduction in power consumption, and the spatial correlation estimation method returns a solution that is close to that returned by the exact simulation method at a fraction of the computational cost.

Table 3. Power Consumption Reduction Results for Berger-code Checkers for some Combinational Circuits when the *minimal.genlib* library is used

Circuit			Average Power (100 random orderings)	Exact Simulation Method		Spatial Correlation Estimation Method	
Name	Num PIs	Num Pos		Power	Runtime (seconds)	Power	Runtime (seconds)
x2	10	7	268	230	3510	250	16
cu	14	11	267	210	19140	210	31
sct	19	15	1234	1101	22510	1134	53
b9	41	21	2405	1987	77400	2161	96
seq	41	35	2501	*	*	1765	244
k2	45	45	3234	*	*	2668	396

Our implementation decomposes the checker using 2-input gates when building the reduced cost function. If correlation values for signals considered three at a time were available, it

would be possible to write the transition probabilities for some more gates when building the reduced cost function. This can improve the accuracy of the proposed procedure, and better results could be obtained by decomposing the checker to up to 3-input gates. Our experimental results indicate that it is sufficient to consider correlations between pairs of signals when building the reduced cost function.

5. Conclusion

As CED increasingly becomes a necessity in mainstream commercial electronics, there is an urgent need for techniques to reduce the associated overhead costs. In this paper, we have presented a novel approach for reducing the power consumption in checkers used for CED. The method is applicable to any functionally symmetric checker. It analyzes spatial correlations between the outputs of the circuit that drives the checker to order them such that switching activity (and hence power consumption) in the checker is minimized. The reduction in power consumption comes at no additional impact to area or performance and does not require any alteration to the design flow. The only cost is the time for computing the input ordering for the checker that minimizes the power consumption. The methodology can be easily integrated into existing CAD tools.

References

[Cohen 99] Cohen, N., Sriram, T. S., Leland, N., Moyer, D., Butler, S., and Flatley, R., "Soft error considerations for deep-submicron CMOS circuit applications," *Intl. Electron Devices Meeting Technical Digest,* 1999.

[Kavousianos 98] Kavousianos, X., and D. Nikolos, "Novel Single and Double Output TSC Berger Code Checkers," *Proc. VLSI Test Symposium,* pp. 348-353, 1998.

[Kirkpatrick 83] Kirkpatrick, S., Gelatt, C. D., and Vecchi Jr., M. P., "Optimization by Simulated Annealing," *Science*, pp. 671-680, May. 1983.

[Metra 96] Metra, C., Favalli, M., and Ricco, B., "Tree checkers for applications with low power-delay requirements," *Proc. IEEE Intl. Symposium on Defect and Fault Tolerance in VLSI Systems,* pp. 213-220, 1996.

[Najm 94] Najm, F. N., "A Survey of Power Estimation Techniques in VLSI Circuits," *IEEE Trans. Very Large Scale Integration,* Vol. 2, No. 4, pp. 446-455, Dec. 1994.

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

[Piestrak 01] Piestrak, S. J., Bakalis, D., and Kavousianos, X., "On the Design of Self-Testing Checkers for Modified Berger Codes," *Proc. IEEE Intl. Online Testing Workshop,* pp. 153-157, 2001.

[Pradhan 86] Pradhan, D. K., Chapter 5, *Fault Tolerant Computing – Theory and Techniques,* Vol. 1, Prentice-Hall, NJ, 1986.

[Prasad 96] Prasad, S. C., and Roy, K., "Transistor Reordering for Power Minimization Under Delay Constraint," *ACM Trans. Design Automation of Electrical Systems,* Vol. 1, No. 2, pp. 280-300, Apr. 1996.

[Schneider 96] Schneider, P. H., and Krishnamoorthy, S., "Effects of Correlations on Accuracy of Power Analysis – An Experimental Study," *Proc. Intl. Symposium on Low Power Electronics and Design (ISLPED),* pp. 113-116, 1996.

[Sentovich 92] Sentovich, E. M., Singh, K. J., Lavagno, L., Moon, C., Murgai, R., Saldanha, A., Stephan, P. R., Brayton, R. K., and Sangiovanni-Vincentelli, A. L., "SIS: A System for Sequential Circuit Synthesis," Technical Report Memorandum No. UCB/ERL M92/41, University of California, Berkeley, 1992.

[Tan 94] Tan, C., and Allen, J., "Minimization of power in VLSI circuits using transistor sizing, input ordering, and statistical power estimation," *Proc. Intl. Workshop on Low Power Design,* pp. 75–80, Apr. 1994.

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

[Ziegler 96] Ziegler, J. F., *et al.*, "IBM experiments in soft fails in computer electronics," *IBM Journal of Research and Development*, Vol. 40, No. 1, Jan. 1996.