On the design of self-checking functional units based on Shannon circuits

Michele Favalli DI - University of Ferrara - Via Saragat, n. 1 - Ferrara (Italy)

Cecilia Metra

DEIS - University of Bologna - Viale Risorgimento, n. 2 - Bologna (Italy)

Abstract

This paper investigates the application of Shannon (BDD) circuits, that feature interesting low-power capabilities, to the design of self-checking functional units. A technique is proposed that, by using a time redundancy approach, makes this kind of circuits totally self-checking with respect to stuck-at faults. For a set of possibly used pass-transistor-based CMOS implementations, we show that the totally self-checking or the strongly fault secure properties hold for a wider set of realistic faults, including transistors stuck-open/on and bridgings.

1. Introduction

Self-checking circuits [8] are widely used in safety critical digital systems to detect on-line the presence of errors immediately after their occurrence. Therefore, they are a basic building block of fault tolerant systems [22] making use of error recovery procedures and of fail-safe systems [20]. Self-checking circuits consist of a functional unit and a checker. The functional unit outputs are encoded by means of an error detecting code and are continuously verified by the checker.

The main design target for self-checking circuits is to avoid that an incorrect data is produced at the functional unit's outputs without an error indication given by the checker. This can be achieved under defined fault hypotheses. In particular, it should never occur that the functional unit gives an incorrect codeword because of a fault. In fact, the checker would be not able to detect such an error. The totally self-checking [1] and the strongly fault-secure properties [23] ensure this kind of behavior.

In practice, the constraints on the design of the functional unit prevents from the full exploitation of logic minimization techniques. In fact, it should be avoided that the effects of a fault affecting a node reach different circuit outputs through inverting and non-inverting paths, thus resulting in errors not detectable by the error codes used in VLSI. To solve this problem, systematic design approaches have been developed which refers to two-level or multilevel design styles [10, 11, 13, 19, 21, 9, 6, 26, 3, 2].

The optimal synthesis of self-checking functional units for any kind of code, however, is still an opened problem. The proposed solutions typically require a significant area overhead, that in some cases may be equivalent to that of duplication (see, for instance, the comparison in [2]).

At this regard, it should be noticed that, in addition to the commonly used multilevel combinational design style based on libraries of static CMOS gates, other design styles are currently explored to match the design requirements of submicron VLSI ICs. In particular, circuits based on the direct mapping of Shannon expansion, as exploited in boolean function representation techniques based on BDDs (binary decision diagrams) [4], are attracting particular attention because of their low-power capabilities [25, 14, 5].

Such capabilities may be required also for self-checking circuits (for instance biomedical systems), where, conversely, speed may be not a critical issue.

Based on these motivations, this paper investigates some properties of Shannon circuits with respect to the implementation of self-checking functional units.

In particular, we present a technique that, by exploiting such properties, makes Shannon circuits totally selfchecking with respect to stuck-at faults. This target is achieved by means of a simple time redundancy of operations, so that there is no need to resynthesize logic functions optimized with respect to area and delay by using error detecting codes.

Functional units designed in this way do not require any modification with respect to conventional Shannon circuits and, therefore, maintain their low-power characteristics [25, 14, 5]. In addition, such circuits can operate in a conventional way.

In this work, we analyze possible implementations of such a kind of circuits based on CMOS pass-transistor multiplexers and we show that that Shannon circuits can be made totally self-checking or strongly fault-secure with respect to non-stuck-at faults, such as transistors stuck-on, stuck-open and bridgings,

This paper is organized as follows. In section 2, Shannon circuits operations are briefly described. In section 3, some basic concepts and definitions of self-checking circuits are introduced. The basic idea of this work is described at the logic level in section 4. In section 5, possible CMOS implementations of the proposed kind of circuits are described and shown totally self-checking with respect to stuck-at faults. In section 6, the self-checking capabilities of such circuits with respect to transistor and bridging faults are analyzed.

2. Shannon circuits

The Shannon theorem can be used to represent logic functions as Binary Decision Diagrams (BDDs) [4]. Several versions of BDDs exist that use different strategies to achieve a compact representation of the logic function.

BDDs have been widely used in logic synthesis, verification and test, because their reduced order version (ROB-DDs) is a canonical form with ease of function manipulation. Since each BDD node corresponds to a 2-way multiplexer, BDDs individuate a logic network implementing the represented function (Fig. 1).



Figure 1. A BDD and the corresponding implementation based on multiplexers.

This kind of structure takes advantage from the compactness of CMOS multiplexers implemented by means of pass transistors. Moreover, only a single path is sensitized at a time between the 1 and 0 constant inputs and the output (Fig. 2).

Single path sensitization and general properties of passtransistor logic give to Shannon circuits low-power characteristics. The property of most interest to this work is the single path sensitization. Thus, the proposed self-checking technique applies to a wide class of BDD based representations and to the corresponding circuits. However, in all following examples we will consider ROBDDs as a refer-



Figure 2. Example of sensitized path in a Shannon circuit.

ence.

3. Self-checking circuits

Let us review some properties that self-checking functional units should satisfy in order to correctly perform their operations, in particular, to guarantee that an incorrect codeword can never be produced because of a fault.

Such properties are based on the following fault hypothesis:

- faults occur one at time;
- the time elapsing between the occurrence of two subsequent faults is long enough to allow the application of all input vectors to the considered circuit.

A circuit is fault-secure for a set of faults F, if for every fault in F, the circuit never produces an incorrect codeword at the output for an input codeword.

A circuit is self-testing for a set of faults F, if for every fault in F, the circuit produces a non codeword at the output for at least an input codeword.

If a circuit is both fault secure and self-testing it is said to be totally self-checking [1].

A circuit, instead, is said to be strongly fault-secure [23] with respect to a set of faults F if for every fault in F, either:

- a) the circuit is self-testing or
- b) the circuit is fault secure, and if another fault from F occurs in the circuit then either property (a) or (b) is true for the fault sequence.

4. Self-checking Shannon circuits

In this work, Shannon circuits are made self-checking by employing a time redundancy approach. In particular, the two constants 1 and 0 are switched in a clock period (for instance, in the first semiperiod they have their nominal value, while they are complemented in the subsequent semiperiod). If the input signals are constant in a such period, the output will assume a logic value in a semiperiod and its complement in the other one (Fig. 3). Therefore, if the circuit output switches in a period, it is interpreted as error free, while if it remains constant, it is interpreted as erroneous. In practice, the output value in the two frames constitutes a time domain one variable two-rail code. Hence, such a code can be easily checked by using for each output line two flip-flops and a two-rail checker.

Other approaches exploiting a temporal coding of signals have been used in the design of functional units are described in [21] (where, however, additional hardware is required to implement a dual replica of the circuit) and in [7] (where clocked CMOS gates are used).



Figure 3. Behavior of the proposed self-checking Shannon circuit in a period.

Now, let us demonstrate that a functional unit designed in this way is totally self-checking with respect to stuck-at faults (i.e. it satisfies the fault-secure and the self-testing properties). To this purpose we will use a logical model of BDD nodes (in practice corresponding to a SP implementation of multiplexers), while detailed electrical level considerations will be introduced in the next section. The multiplexer is considered as a component implementing the logic function out = as + bs' (where, a and b are data inputs, and s is the selector input).

We have three different categories of multiplexer faults: i) s-at-0/1 on the output; ii) s-at-0/1 on the selector input; iii) s-at-0/1 on the data inputs.

In case i) it is evident that, once the fault is activated and its effects are made observable at the functional unit outputs (i.e. the multiplexer is on the sensitized path), it gives rise to a constant output, because the same path is selected by the circuit inputs in the two semiperiods. The same holds in case iii).

Case ii) is slightly more complex. Consider a fault affecting s (the s' case is dual). The multiplexer function becomes: a) a + bs' in the s-at-1 case; b) bs' in the stuck-at 0 case. The fault is activated with: s = 0 (case a)) and s = 1 (case b)), respectively. Therefore, in case a), the multiplexer

performs the function a + b, and in case b) the function 0. The case (a) is the most interesting: if the signals a and b have the same value, the fault-free value will be present at the circuit output, otherwise the constant value 1 will be produced and recognized as a logic error.

Notice that, if the signal s' is generated by a single inverter for all multiplexers driven by the same input signal, the same considerations can be made for the stuck-at faults on the output of this inverter (because only a single path is sensitized).

In addition, it is easy to verify that also stuck-at faults affecting the constant inputs 0 and 1 would result to a constant output when a path between the faulty signal and the output is sensitized.

Therefore, it never occurs that a functional unit output produces a wrong sequence of logic values in the same period (i.e. 01 instead of 10 or viceversa) because of a fault affecting a multiplexer or the constant inputs. Then the Shannon circuit is fault-secure.

Moreover, it should be noticed that the considerations made in this section for a single output circuit hold also for multioutput circuits where the originating BDD has nodes which are shared between more outputs (such circuits in fact still maintain the property of single path sensitization between the constants inputs and each output).

As regards the self-testing property, it should be noticed that we refer to reduced BDDs, where there are no redundant nodes (i.e. nodes were the function of both subtrees is the same, or the node value is never observable at the output). This means that for each multiplexer, at least a configuration exists for which a path is sensitized between the input a and the circuit output. The same hold for the input b. These conditions are present in almost all BDD based circuit implementations, because the presence of redundancy impairs not only testability, but also area optimization.

Under these hypotheses, it is easy to verify that the faults of kind i) and iii) of each multiplexer are testable with the adopted methodology (that ensures to test both s-at-0 and 1 faults, also affecting the multiplexer inputs connected to the constants 0 and 1).

Faults of kind ii) are also testable. Suppose that, under fault-free conditions, the path between the input a of the multiplexer and the circuit output is sensitized, that is s = 1 and s' = 0. If s is stuck-at-0, the multiplexer output is at the logic 0 in both semiperiods. If s is stuck-at-1, the multiplexer function becomes a + b and the fault may be detected if $a \neq b$. If there is no configuration setting up such condition, when the multiplexer output is observable at the circuit output, its inputs have always the same value, that is, such multiplexer is redundant in contrast to the starting hypotheses.

Since all possible stuck-at faults affecting the functional unit are detectable, also the self-testing property is verified.

Therefore, a functional unit implemented as a Shannon circuit and making use of the proposed methodology is totally self-checking with respect to stuck-at faults.

5. CMOS implementation

When using the CMOS technology, pass-transistors allow a very compact implementation of multiplexers. As a consequence, Shannon circuits can be, on principle, implemented as binary trees of pass-transistors [25]. Timing issues, however, may require the insertion of buffers to the purpose of signal restandardization [14]. At this regard, Figs. 4 instantiate three possible implementations of a Shannon circuit.

Under fault-free conditions, the behavior of the passtransistor multiplexer is the same as that described at the logic level, while in the presence of faults, some additional consideration should be made.

As regards stuck-at faults of kind i) and iii), the same considerations made in the previous section hold for both the fault-secure and the self-testing properties.

In case ii), instead, the logic function of the passtransistor multiplexer (Fig. 4a) implies that, in case of a stuck-at-0 fault on s or s', the multiplexer output is in a high impedance state when the fault is activated, thus retaining the value it had before activation for both semiperiods. Therefore, whenever a path is sensitized through such multiplexer, such faults give rise to a constant output. In the case of a stuck-at 1 on one of such signals, instead, when the fault is activated, both transistors are ON, and two possibilities are in order depending on the values of a and b. If a = b, the circuit output has the fault-free value. If $a \neq b$, instead, there is a conflict between the different values driven by signals a and b, that results in an intermediate voltage at the multiplexer output, whose value depends on the conductance of the conflicting networks. We should avoid that one of the two paths (that driving a and that driving b) prevails over the other. In practice, the logic 0 or 1value should always prevail in order to generate a constant value. Otherwise, an intermediate value can be generated for both semiperiods.

In case all multiplexers are buffered (Fig. 4b), this kind of behavior can be simply achieved. In fact, the use of *n*channel pass-transistors makes low logic values always win the conductance conflict.

In case no buffer is used, such as in [25] (Fig. 4a), instead, some problem may arise when networks with a small number of series transistors are in conflict with networks with a large number of series transistors. In such cases, the value driven by the shortest path always prevails and two different intermediate voltage values are propagated to the circuit output. In this case, the output circuitry should be made able to detect non valid logic values, such as the circuit in [12]. The case where buffers are placed in specific positions inside the circuit (Fig. 4c) is slightly more complex and pose some constraint on the circuit design. The presence of buffers, in fact, does not allow the propagation of intermediate values to the circuit output. Hence, each buffer should interpret each pair of intermediate voltages (present at its input during the two semiperiods) in the same way. Let V_L and V_H be the minimum and maximum values of intermediate voltage originated at the buffer input because of internal faults of kind ii). To ensure a correct fault secure behavior (i.e. to avoid wrong alternating outputs) it is sufficient that the logic threshold of the buffer (V_{LT}) is $V_{LT} > V_H$ or $V_{LT} < V_L$.

The design space to be explored to satisfy this condition involves the buffer and pass-transistor sizing, and the difference between the shortest and the longest path in the subtree feeding the buffer input.

In particular, we have supposed to have symmetric buffers (with $V_{LT} = V_{DD}/2$) and equally sized passtransistors. In such case, by using a $0.5\mu m$, 3.3V power supply technology, we have a maximum tolerable difference of 2 transistors between different paths feeding the same buffer. This ensures a noise margin of 0.3V with respect to circuit parameter variations of 15% with respect to their nominal value. Of course, variable transistor sizing would increase the tolerable difference in the transistor number, but it would also increase the design complexity.

Therefore, we have shown that all the considered CMOS multiplexer implementations are fault-secure with respect to stuck-at faults. As regards, the self-testing property, the considerations made in section 4 apply directly to pass-transistor implementations. Therefore, the CMOS implementation of Shannon circuits is totally self-checking with respect to stuck-at faults.

Finally, let us remark that the proposed design methodology can be used with any kind of multiplexer cells possibly different from those analyzed in this section.

6. Non stuck-at faults

In this section, we analyze the self-checking properties of Shannon circuits with respect to typical CMOS faults, such as transistor stuck-on, stuck-open and bridging faults.

6.1 Transistor stuck-on/open

Transistor stuck-on/open faults affecting the passtransistors are equivalent to stuck-at faults on the selection signals that have been considered in the previous section.

As regards the faults affecting buffers, transistor stuckopens can be easily detected because the buffer output is in a high impedance state when the fault is activated thus retaining its previous value. Therefore, if a path is sensitized through such gate, the circuit output maintains a constant value along the two semiperiods of interest.



Figure 4. Different implementations of the same Shannon circuit based on: a) pass-transistor multiplexers; b) buffered pass-transistor multiplexer; c) pass-transistor multiplexers with inserted speed-up buffers.

Transistors stuck-on, instead, originate intermediate faulty voltages at the buffer output. Consider, for instance, a stuck-on fault on the buffer pull-down transistor. Such a fault is activated by a logic 0 at the buffer input, hence, the input fault-free sequences 01, 10 result in the following sequence of output voltages $V_X 0$, $0V_X$, where V_X is the faulty intermediate voltage. If V_X is propagated as a logic 1, the correct sequence is present at the circuit output, otherwise a constant low value is produced. It is evident that the fault secure property is always verified, while only in the second case, the fault cannot be detected.

In this case, it can be verified that the circuit is strongly fault-secure.

6.2 Bridging faults

Bridging faults have been recognized as one of the most common cause of failures in CMOS circuits [17]. Methodologies have been developed for the design of functional units that are self-checking with respect to bridging faults [18]. In addition, self-checking built-in current sensors (BICs) have been developed that are capable to detect online the presence of such kind of faults by revealing the presence of abnormal I_{DDQ} current. The presence of such sensors may be required by low-power applications because it may be necessary to turn-off devices with abnormal current assumption. In such cases, the BICs can be used to complement with the checker operations [16]. If BICs are not used, bridging faults should be analyzed from the point of view of functional effects.

In this section we discuss with some detail the detection of bridging faults in the considered kind of pass-transistor logic. In this work, we consider resistive bridging faults between two nodes of the circuit. In particular, we can have bridgings between:

- a) two internal nodes;
- b) an internal node and a selection node;
- c) two selection nodes.

In this case, the results strongly depends on the electrical level implementation of the circuit. Therefore the three cases of Fig. 4 should be considered separately. General qualitative considerations will be made and electrical level simulation will be applied to an example, to account for the parametric characteristics of such kind of faults.

Let us first consider the circuit implementation without buffers. Let us also suppose that the selection signals are global.

Bridging faults of kind a) are activated and made observable if one of the two involved nodes is on the sensitized path, and the other is not on such a path and has a different logic value. Under such conditions, intermediate logic values are present at the faulty nodes in both semiperiods and are propagated to the circuit output, where they can be recognized similarly to the fault effects of stuck-ats of kind ii). All other conditions result in the fault-free output, therefore the circuit is fault secure with respect to such faults. If the circuit is non redundant, it exists at least one input vector activating the fault. Hence, the circuit is also self-testing and, therefore, totally self-checking with respect to bridging faults of kind a).

In the case of faults of kind b), the hypotheses on the selection signal imply that they are driven by large buffers.

Hence, it can be reasonably supposed that such signals prevail over those internal to the circuit when the fault is activated. Fault effects can be made observable at circuit output if the internal node is on a sensitized path. Therefore, a constant output is present at the output of the circuit, because the selection signal does not change in a period.

The possible conditions set by faults of kind of c) are complex: the two involved signals assume intermediate voltages, hence some pass-transistor may be partially ON and the fault effects may propagate also through the inverter generating the complement of an involved selection signal. In this case, in order to avoid the presence of undetectable errors, it is important that the inverters generating the complement of the input signals present a low logic threshold. Therefore, intermediate voltages at their inputs are interpreted as high, thus producing an output erroneous low value. This results in erroneously OFF pass-transistors, and avoids that, because of an erroneously ON pass-transistor, a wrong path is activated.

Consider, for instance, the example of Fig. 5, where in the fault-free circuit it is a = 0, b = 1 and c = 0. Because of the bridging (here supposed with a very low resistance), it is V(a) = V(b) = 1.65V. Suppose that the inverters ga and gb interpret such a value as low, so that it is V(a') = V(b') = 2.3V. In this case, a wrong path (stronger than the fault-free one) is activated, producing, in the two semiperiods, two intermediate voltages at the output, that may be erroneously recognized as correct. If the inverters have a low logic threshold, instead, V(a') = V(b') = 1.3V, thus (weakly) activating only the correct path in the two semiperiods.

Therefore, also for this kind of faults, the fault-secure property is guaranteed. It can be easily verified that also the self-testing property is verified, hence the circuit is totally self-checking with respect to bridging faults of kind c).

In the case of buffered multiplexers, bridging faults of kind a) may give rise to an output error if at least one of the two involved nodes is on the activated path. In this case, unfortunately, a bridging between an inverter output and a inverter input may result in an undetectable error. In fact, the inverter output prevails over the conflicting network, which includes a pass-transistor.

Therefore, the fault-secure property is not verified for a subset of faults of kind a). These faults can be avoided by acting at the layout level. This is possible because one node is internal to a multiplexer cell, and it has been reported that bridgings between external signals of cells are the most likely [24]. As an alternative, a different kind of buffers can be used. The idea is very simple: two buffers with different logic thresholds are placed in parallel Fig. 6. In the presence of an intermediate voltage at their inputs, the inverter with low logic threshold provides a low output, while that with a high threshold gives rise to an high output.



Figure 5. Example of problems due to bridging faults of kind c). The figure shows the fault-free path (a) and that additionally sensitized under faulty conditions (b).

buffer output is an intermediate voltage that is propagated to the circuit output. Notice that the same considerations hold also in the case of Shannon circuits with inserted buffers. In practice, such kind of buffers behave as conventional buffers in the presence of fault-free values, while they allow the propagation of faulty intermediate voltages.



Figure 6. Buffer used to ensure the propagation of faulty intermediate voltages.

Considerations, similar to those made in the case of the circuit without buffers, hold for bridging faults of kind b) and c). Therefore, with the use of suitable buffers, also Shannon circuits with buffers can be made totally self-checking with respect to resistive bridging faults.

impl.	$R = 100\Omega$	$R = 1000\Omega$	$R = 5000\Omega$
(a)	97.97%	100.0%	100.0%
(b)	91.04%	100.0%	57.22%
(c)	92.21%	93.06%	79.7%

Table 1. Percentage of resistive bridging faults for which the totally self-checking property holds. The three possible implementations of the benchmark cm82a have been considered: a) pass-transistor multiplexers; b) buffered pass-transistor multiplexer; c) pass-transistor multiplexers with inserted speed-up buffers.

6.2.1 Bridging fault simulation

In order to analyze the effects of (resistive) bridging faults, a simple circuit (**cm82a**) taken from the *mcnc* benchmark set [15] has been used to design self-checking Shannon circuits. The circuit has 5 inputs and 3 outputs and features 15 multiplexers. In particular, three different versions of **cm82a** have been implemented, each of them corresponding to one of the possible buffering alternatives considered in this work. Such circuits have been simulated at the electrical level for all possible bridging faults (including feedback bridging faults) for different values of the bridging resistance (R). Simulation results are shown in Tab. 1.

In the circuit implementation without buffers, results show that the circuit is totally self-checking for a very large fraction of faults (without any need for the additional output circuitry used to detect intermediate voltages), while it can be made totally self-checking for the other faults by adding the circuit capable of recognizing intermediate output voltages.

In the cases where pass-transistor multiplexers are all or in part buffered, the achieved results (Tab. 1) show that, by simply using suitably sized conventional inverters, the totally self-checking property is verified for good percentages of bridging faults for low values of R. In the case where all multiplexers are buffered, such percentage decreases dramatically for $R = 5000\Omega$. This is due to bridging faults that do not satisfy the self-testing property because they are too resistive to alter the circuit functionality (i.e. they never produce wrong codewords and, therefore, further simulations are required to verify the fault-secure property). Of course, by using the circuit of Fig. 6, the totally self-checking property can be verified also for the remaining bridgings.

7. Conclusions

This paper investigates the use of Shannon based circuits for the implementation of self-checking functional units. This kind of circuits can be made self-checking without the need to bring modifications (typically resulting in additional hardware) to the design of the functional unit. The performed analysis show that the totally self-checking goal can be achieved for a wide set of realistic faults. Some problem has yet to be solved, because some function cannot be efficiently mapped on a BDD without partitioning. Therefore, the presented technique has to be extended to partitioned circuits. In addition, we are investigating its application to the design of checkers.

References

- D. A. Anderson and G. Metze. Design of Totally Self-Checking Circuits for m-out-of-n Codes. *IEEE Trans. on Computers*, C-22:263 – 269, Mar. 1973.
- [2] C. Bolchini, F. Salice, and D. Sciuto. Designing networks with error detectio properties through the fault error relation. In *Symp. on Defect and Fault Tolerance in VLSI Systems*, pages 290 – 297, 1997.
- [3] C. Bolchini, F. Salice, and D. Sciuto. A novel methodology for designing TSC combinational networks based on the parity bit code. In *Proc. of IEEE Eur. Design and Test Conf.*, pages 440 – 444, 1997.
- [4] R. E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Trans. on Computers*, C35: 677–691, 1986.
- [5] P. Buch, A. Narayan, A. Newton, and A. Sangiovanni-Vincentelli. Logic synthesis for large pass transistor circuits. In *Proc. of IEEE Int. Conf. On Computer Aided Design*, pages 663 – 670, 1997.
- [6] F. Busaba and P. Lala. Self-checking combinational circuit design for single and unidirectional multibit error. J. of Electronic Testing Theory and Applic., (5):19 – 28, 1994.
- [7] B. R. C. Metra, M. Favalli. Signal coding technique and CMOS gates for strongly fault-secure combinational functional blocks. In *IEEE International Symposium on Defect* and Fault Tolerance in VLSI Systems, pages 174–182, 1998.
- [8] W. C. Carter and P. R. Schneider. Design of dynamically checked computers. In *Proc. IFIP '68, Edinburgh, Scotland*, pages 878 – 883, 1968.
- [9] K. De, C. Natarajan, D. Nair, and P. Banerjee. RSYN: asystem for automated synthesis of reliable multilevel circuits. *IEEE Trans. on VLSI*, 2(2):186 – 195, 1994.
- [10] M. Diaz, P. Azema, and J. M. Ayache. Unified Design of Self-Checking and Fail-Safe Combinational Circuits and Sequential Machines. *IEEE Trans. Comput.*, C-28:276 – 281, March 1979.
- [11] F. Salice, M. Sami and D. Sciuto, Synthesis of multi-level self-checking logic. In Proc. of IEEE Int. Work. on Defect and Fault Tolerance in VLSI Systems, pages 115 – 123, 1994.
- [12] M. Favalli, B. Riccò, and L. Penza. A novel DFT technique for critical bridging faults in CMOS and BiCMOS ICs. In *IEEE European Design and Test Conference*, pages 568 – 572, 1995.
- [13] N. K. Jha and S.-J. Wang. Design and Synthesis of Self-Checking VLSI Circuits. *IEEE Trans. on CAD*, 12:878 – 887, June 1993.
- [14] K. Konishi and et al. A logic synthesis system for the passtransistor logic SPL. In SASIMI, pages 32 – 39, 1996.

- [15] R. Lisanke. Logic synthesis and optimization benchmarks user guide v. 2.0 - technical report. Microelectronics Center of North Carolina.
- [16] J.-C. Lo, J. C. Daly, and M. Nicolaidis. Design of Static CMOS Self-Checking Circuits using Built-In Current Sensing. In *Proc. of Int. Symp. Fault-Tolerant Comput.*, pages 104 – 111, 1992.
- [17] W. Maly. Realistic Fault Modeling for VLSI Testing. In Proc. of Design Automation Conf., pages 173 – 180, 1987.
- [18] C. Metra, M. Favalli, P. Olivo, and B. Riccò. On-Line Detection of Bridging and Delay Faults in Functional Blocks of CMOS Self-Checking Circuits. *IEEE Trans. on CAD*, 16(7):770–776, July 1997.
- [19] T. Nanya and M. Uchida. The Design of Strongly Fault-Secure and Strongly Code-Disjoint Combinational Circuits. In *Proc. of Joint Fault-Tolerant Computing Symposium*, pages 245 – 250, 1989.
- [20] M. Nicolaidis. Fail-safe interfaces for VLSI: theoretical foundations and implementation. *IEEE Trans. on Comput*ers, 47(1):62 – 77, 1998.

- [21] V. Saposhnikov et al.. Self-dual parity checking a new method for on-line testing. In *Proc. of IEEE VLSI Test Symp.*, pages 162 – 168, 1996.
- [22] D. Siewiorek. Architecture of fault-tolerant computers: An historical perspective. *Proc. of the IEEE*, 79(12):1710 – 1734, 1991.
- [23] J. E. Smith and G. Metze. Strongly Fault Secure Logic Networks. *IEEE Trans. on Computers*, C-27:491 – 499, June 1978.
- [24] J. Sousa, F. Goncalves, and J. Teixeira. IC defect based testability analysis. In *Proc. of IEEE Int. Test Conf.*, pages 500 – 509, 1991.
- [25] M. Tachibana. Heuristic algorithms for FBDD node minimization with application to pass-transistor-logic and DCVS synthesis. In SASIMI, pages 96 – 101, 1996.
- [26] N. A. Touba and E. J. McCluskey. Logic synthesis techniques for reduced area implementation of multilevel circuits with concurrent error detection. In *Proc. of IEEE Int. Conf. On Computer Aided Design*, pages 651 – 654, 1994.