VLSI DESIGN
1998, Vol. 5, No. 4, pp. 333–345
Reprints available directly from the publisher
Photocopying permitted by license only

Self-Checking Combinational Circuits with Unidirectionally Independent Outputs

A. MOROSOW[a,*], V. V. SAPOSHNIKOV[b], VL. V. SAPOSHNIKOV[b] and M. GOESSEL[a]

[a] *Department of Computer Science, Fault-Tolerant Group at the University Potsdam, P.O. Box 601553, Potsdam 14415, Germany;*
[b] *Railway-Transportation University of St. Petersburg, pr. Moskovskij 9, Sankt-Petersburg, Russia*

In this paper we propose a structure dependent method for the systematic design of a self-checking circuit which is well adapted to the fault model of single gate faults and which can be used in test mode.

According to the fault model considered, maximal groups of independent and unidirectionally independent outputs of an arbitrarily given combinational circuit are determined. A parity bit is added to every group of independent outputs. A few additional outputs are added to every group of unidirectionally independent outputs. In the error free case, these groups of unidirectional independent outputs together with their corresponding additional outputs are elements of a unidirectional error detecting code; for example, a Berger code or an r-out-of-s code.

It is demonstrated how the pairs of (unidirectionally) independent outputs of a given circuit can be determined. A simple heuristic solution for this problem based on a modified circuit graph is also given.

The maximal classes of (unidirectionally) independent outputs can be computed as cliques of a dependency graph where the nodes of the graph are the outputs of the circuit. The applicability of the proposed method is demonstrated for the MCNC benchmarks circuits.

Keywords: Self-Checking circuits, Independent and unidirectionally independent outputs, Parity-code, Berger-code

1. INTRODUCTION

The design of self-checking circuits[1–3] and the combination of methods for on-line error detection or concurrent checking and testing is of great interest now (s.e.g.[4]). It was first proposed in[5] that on-line error detection by duplication and comparison be combined with testing to use the same hardware in different operational modes. These ideas were further developed in[6,7] for an arbitrary prediction function and for independent outputs in[8]. Parity prediction for mutually disjoint groups of outputs of combinational circuits is a special case of this approach[9,10]. These methods, however, are generally not well-adapted to the specific structure of the monitored circuit and the technical fault model. A method for the design of self-checking and self-testing circuits which is to some extent adapted to the fault model of single gate faults, is given in[8]. In that paper, independent

* Corresponding author.

outputs of the monitored combinational circuit are the groups of outputs for which the parity is computed, duplicated, inverted, and compared with the parities of the original circuit. In normal operation mode, every single gate fault forcing an output to be erroneous for the first time will be detected. As well, every single gate fault will be detected in test mode. In reality, however, there are only very few cases where all the outputs of a circuit can be combined into groups of independent outputs of a reasonable size which are implemented by completely different gates.

In[11], weakly independent outputs are used instead of independent outputs for the design of self-testing circuits. Low cost implementations of self-testing circuits are very often possible, but in on-line mode faults may only be detected with some degree of latency and the circuits are not self-checking.

As a generalization of[8], we propose in this paper a structure dependent method for the systematic design of self-checking combinational circuits which is well-adapted to the fault model of single gate faults and the structure of the considered circuit. Groups of independent outputs as well as groups of unidirectionally independent outputs are determined. Two outputs are unidirectionally independent with respect to a given fault if, in the presence of this fault either both outputs are correct, only one output is erroneous, or both the outputs are unidirectionally erroneous.

In normal operation mode, the proposed circuits are self-checking with respect to all single gate faults. In test mode, these error detection circuits guarantee a 100% fault coverage for all nonredundant single stuck-at- 0/1 faults.

In this paper, a given combinational circuit is considered. In[12] a given circuit is modified by use of a special input (output) encoding. The proposed encoding technique guarantees that all internal single stuck-at faults as well as all single stuck-at faults of the input lines will result either in single bit errors or in unidirectional multibit errors of the outputs of the modified circuit. The rest of the paper is organized as follows.

In section 2, the basic notions and notations such as unidirectionally independent outputs, maximal groups of unidirectionally outputs, and generalized circuit graph are given. Basic theorems useful for the determination of independent outputs and unidirectionally independent outputs are also presented. In section 3, it is shown how independent and unidirectionally independent outputs can be used to design self-checking circuits. A few additional outputs are added to every group of unidirectionally independent outputs. In the error free case, these groups of unidirectionally independent outputs are elements of an error detecting code which detects all unidirectional errors. Every group of independent outputs is checked by a parity bit. In section 4, it is explained how pairs of (unidirectionally) independent outputs can actually be determined. The determination of maximal groups of (unidirectionally) independent outputs can be reduced to the standard graph-theoretical problem of the determination of maximal cliques of a dependency graph, where the nodes of the graph are the outputs of the circuit considered.

In section 5 experimental results for benchmark circuits are given.

2. BASIC NOTIONS AND NOTATIONS

In this paper we consider a combinational circuit f_c with n binary inputs $x_1, \ldots, x_n$ and m binary outputs $y_1 = f_1(x_1, \ldots, x_n), \ldots, y_m = f_m(x_1, \ldots, x_n)$, where $f_1, \ldots, f_m$ are n-ary Boolean functions. Let $x = (x_1, \ldots, x_n)$, and $y = (y_1, \ldots, y_m)$.

The set of technical faults considered is denoted by $\phi = \{\phi_0, \phi_1, \ldots, \phi_K\}$, where ϕ_0 denotes the absence of a fault.

The output y_i under input x in the presence of a fault $\phi_j \in \phi$ is described by

$$y_{i,j} = f_i(\phi_j, x).$$

Then, for the correct output in the absence of a fault, we have

$$y_{i,o} = y_i = f_i(\phi_0, x) = f_i(x).$$

The notion of unidirectionally independent outputs is introduced in the following definition.

DEFINITION 1 The outputs y_i and y_k are called unidirectionally independent with respect to the fault $\phi_j \in \phi$ and with respect to a subset $\mathbf{X} \subseteq X$ of the input set X if we have, for $x \in \mathbf{X}$, either

1. $f_i(x) = f_i(\phi_j, x)$ and $f_k(x) = f_k(\phi_j, x)$

or

2. $(f_i(x) = f_i(\phi_j, x)$ and $f_k(x) \neq f_k(\phi_j, x))$ or $(f_i(x) \neq f_i(\phi_j, x)$ and $f_k(x) = f_k(\phi_j, x))$

or

3. $f_i(x) \neq f_i(\phi_j, x)$ and $f_k(x) \neq f_k(\phi_j, x)$ and $f_i(x) = f_k(x)$.

According to this definition, two outputs are unidirectionally independent if they are both correct, if only one of them is erroneous, or if they are both unidirectionally erroneous. In the last case, both outputs are changing from 0 to 1 or from 1 to 0.

Independent outputs are a special case of unidirectionally independent outputs. Outputs are independent if only the conditions 1 and 2 of Definition 1 are fulfilled. Thus two outputs are independent if they are either both correct or if only one of them is erroneous at a time.

In the following we sometimes omit the subset **X**, especially if this subset is not specified. Now we generalize the definition of a pair of unidirectionally independent outputs with respect to a set of faults and a set of outputs.

DEFINITION 2 The outputs y_i and y_k are called unidirectionally independent with respect to a set ϕ of faults if these outputs are unidirectionally independent with respect to every fault $\phi_j \in \phi$.

DEFINITION 3 The outputs $y_{i_1}, \ldots, y_{i_r}$ form a group of unidirectionally independent outputs if every pair of these outputs is unidirectionally independent.

Similar definitions can be given for independent outputs. Independent outputs can be determined by use of the generalized circuit graph introduced in[9]. In a similar way, unidirectionally independent outputs can be determined by a modification of this generalized circuit graph. This will be explained now.

A combinational circuit C can be described by the connections between outputs and inputs of its logical gates. The gates and the direct outputs of the circuit are called "elements" of the circuit. These elements can be combined into maximal classes C_i of elements with one output. The maximal classes C_i are identified with the nodes N_i of the generalized circuit graph G. A node N_1 of this graph G is connected with a node N_2 by an directed edge directed from N_1 to N_2 if the output of the class C_1 is connected to one of the inputs of the gates of the class C_2. The maximal classes of elements with one output corresponding to the nodes of the generalized circuit graph can be determined as follows:

1. Create a subset for each non-fanout output of the circuit, and mark each element.
2. If the output of an element is connected only to marked elements of a single, already existing subset, add the element to the subset and mark the element.
3. If the output of an element is connected only to already marked elements not all belonging to the same subset mark this element and create a new subset containing this element. The element is the output of the newly opened subset.
4. Continue until all elements are marked and belong to a subset.

In the next step the circuit graph $G = (N, V)$ will be modified into the modified circuit graph $G' = (N', V')$. In addition to the nodes N of the circuit graph G, a node $n_1, \ldots, n_m$ is assigned to every output $y_1, \ldots, y_m$ of the circuit f_c. For the set N' of nodes of G', we have $N' = N \cup \{n_1, \ldots, n_m\}$. In the modified circuit graph G', the nodes $N_i, N_k \in N$ are connected by a non-marked edge if there is a path from the output element of the node N_i to one of the inputs of the output element of N_k with an even number of inversions. The nodes $N_i, N_k \in N$ are connected by a marked edge if there is a path from the output element of the node N_i to one of the inputs of the output element of N_k with an odd

number of inversions. Inversion of the output of the output element of N_i as well as inversions at the input of the output element of N_k are taken into account. Inversions at an input of the output element of N_i as well as inversions of the output of the output element of N_k are not considered for the marking of the edge from N_i to N_k. Thus, two nodes of the modified graph G' are connected by at most two edges, a marked one and an unmarked one.

As an example we consider the combinational circuit of Fig. 1, which implements the following five Boolean functions:

$$y_1 = (\neg x_1)x_2x_3(\neg x_4) \vee (x_1 \vee x_2)x_3x_4 \vee (\neg x_1)(\neg x_3)x_4$$
$$y_2 = \neg(x_1 \vee x_2) \vee x_1x_3(\neg x_4) \vee x_1(\neg x_3)x_4 \vee (x_1 \vee x_2)(\neg x_3)(\neg x_4)$$
$$y_3 = (x_3 \vee x_4)((\neg x_1)(\neg x_2) \vee (\neg x_1)x_3(\neg x_4))$$
$$y_4 = (\neg x_1)(\neg x_2) \vee x_1(\neg x_3)x_4 \vee x_1x_3(\neg x_4)$$
$$y_5 = \neg(x_3 \vee x_4) \vee (\neg x_1) \vee (\neg x_2).$$

The modified generalized circuit graph is shown in Fig. 2. According to Fig. 2, two edges are drawn

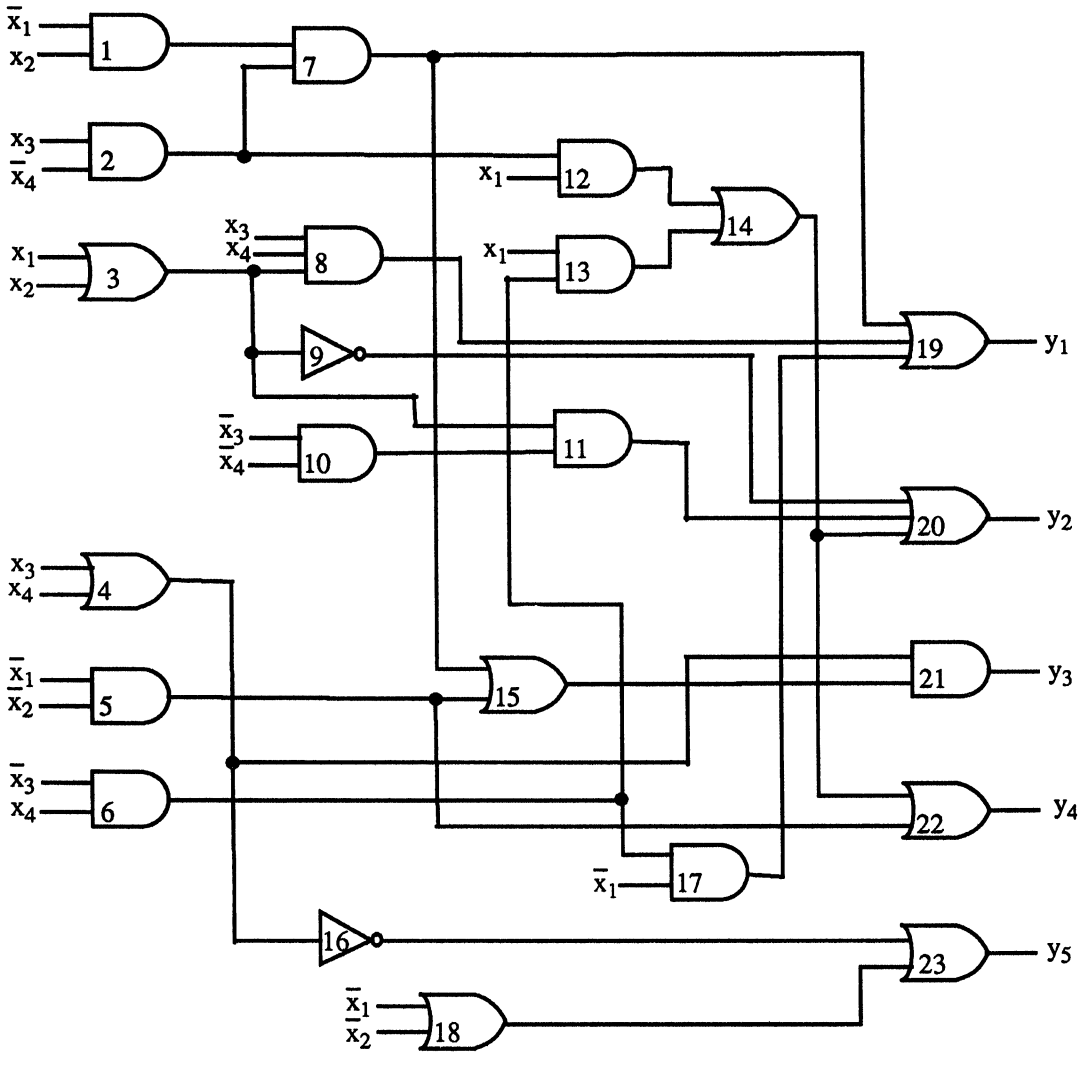

FIGURE 1 Example of a combinational circuit.

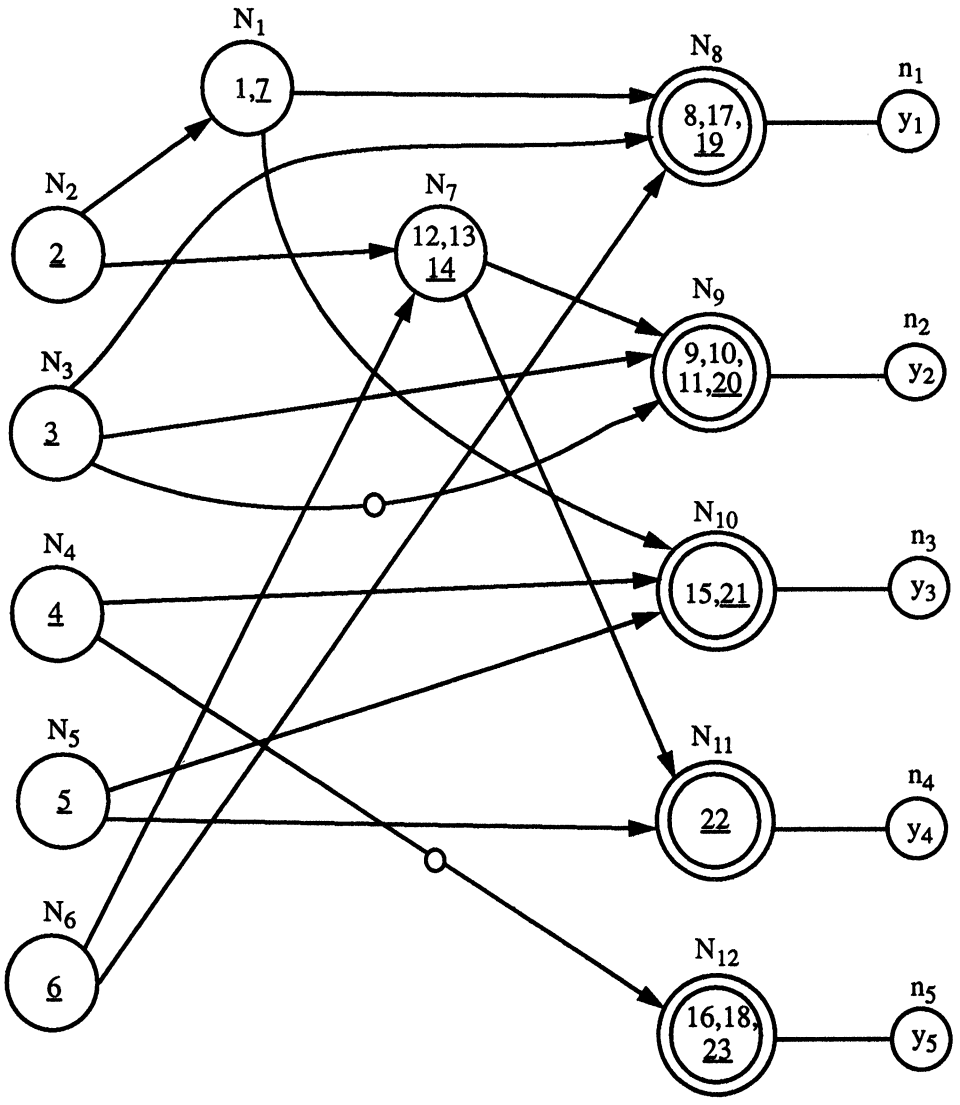

FIGURE 2 Modified generalized circuit graph of the example circuit of Fig. 1.

from the node N_3 to the node N_9. The first edge corresponds to a path with one inversion (an odd number of inversions) from element 3 to element 20 via element 9. The second edge via element 11 has no inversion (an even number of inverversions).

Similarly, as in[9], a node N_i of the modified generalized circuit graph is called structural essential for the node N_k if there is at least one path from N_i to N_k. If there is no path from N_i to N_k the node N_i is called structural non-essential for the node N_k.

The output of the output element e_j^r of the maximal class C_j corresponding to the node N_j will be denoted by z_j. The upper index r of e_j^r is the number of the element in the circuit f_c if the elements of the circuit are numbered.

We are mainly interested in outputs which are (unidirectionally) independent with respect to functional faults of single gates (single gate faults). The maximal classes of elements with one output are very useful notions to describe these faults.

DEFINITION 4 The outputs y_i and y_k are called (unidirectionally) independent with respect to the node N_j if these outputs are (unidirectionally) independent with respect to all single gate faults within the maximal class C_j.

PROPOSITION 1 The outputs y_i and y_k are unidirectionally independent with respect to the node N_j of the generalized circuit graph if and only if we have

$$F_{i,k;j} = \frac{dy_i}{dz_j} \cdot \frac{dy_k}{dz_j}(y_i \oplus y_k) = 0. \qquad (1)$$

Thereby z_j is the output of the output element e_j^r of the maximal class C_j corresponding to the node N_j.

Proof A fault within the class C_j which forces an output y_i to be erroneous for some $x \in X$ necessarily changes the output $z_j(x)$ of C_j into $\neg z_j(x)$, and for every input $x \in X$ there exists a single gate fault within C_j which forces $z_j(x)$ to be $\neg z_j(x)$. (For $z_j(x) = 1$ (0), a stuck-at-0 (1) fault of the output gate of C_j results in an output of 0 (1)).

The inputs of f_c for which a change of the output z_j of C_j results in a change of the output y_s, $s = i, k$, are determined by $dy_s/dz_j(x) = 1$. Thus $dy_i/dz_j(x) \cdot dy_k/dz_j(x) = 1$ determines the inputs for which a change of z_j results in a simultaneous change of both the outputs y_i and y_k, respectively. $y_i(x) \oplus y_k(x) = 1$ determines the inputs for which we have $y_i(x) = \neg y_k(x)$, and $dy_i/dz_j(x) \cdot dy_k/dz_j(x)(y_i(x) \oplus y_k(x)) = 1$ determines the inputs for which a change of z_j simultaneously changes the outputs $y_i(x)$ and $y_k(x)$ in different directions. From these equations we conclude (1).

For independent outputs, we have the following proposition.

PROPOSITION 1′ The outputs y_i and y_k are independent with respect to the node N_j of the generalized circuit graph if and only if we have

$$F'_{i,k;j} = \frac{dy_i}{dz_j} \cdot \frac{dy_k}{dz_j} = 0. \qquad (1')$$

For practical applications, the following sufficient conditions for the determination of (unidirectionally) independent outputs with respect to a node N_j of the modified generalized circuit graph are of interest.

CONDITION 2 The outputs y_i and y_k are unidirectionally independent with respect to the node N_j if

1. N_j is structural non-essential for at least one of the output nodes n_i or n_k,
or
2. all the path's from N_j to n_i and from N_j to n_k in the modified generalized Graph G' contain either an even number of marked edges or an odd number of marked edges.

CONDITION 2′ The outputs y_i and y_k are independent with respect to the node N_j if N_j is structural non-essential for at least one of the output nodes n_i and n_k.

Example By use of Condition 2, the outputs y_1 and y_2 can be dependent with respect to the node N_3, and the outputs y_3 and y_5 can be dependent with respect to the node N_4. To decide whether or not they are unidirectionally independent, we compute

$$\frac{dy_1}{dz_3} = x_3x_4, \quad \frac{dy_2}{dz_3} = (\neg x_1)x_3 \vee (\neg x_1)x_4 \vee x_3x_4,$$
$$\frac{dy_3}{dz_4} = (\neg x_1)(\neg x_2) \vee (\neg x_1)x_3(\neg x_4), \quad \frac{dy_5}{dz_4} = x_1x_2.$$

Now we check condition 1:

$$\frac{dy_1}{dz_3} \cdot \frac{dy_2}{dz_3}(y_1 \oplus y_2) = x_3x_4 \neq 0$$
$$\frac{dy_3}{dz_4} \cdot \frac{dy_5}{dz_4}(y_3 \oplus y_5) = 0.$$

Thus the outputs y_3 and y_5 are unidirectionally independent and the outputs y_1 and y_2 are not.

The following theorem is of great practical interest for the determination of unidirectionally independent outputs.

PROPOSITION 3 Two outputs y_i and y_k are (unidirectionally) independent with respect to the set of all single gate faults ϕ if they are (unidirectionally) independent with respect to all nodes N_j of the generalized circuit graph.

3. DESIGN OF SELF-CHECKING CIRCUITS WITH UNIDIRECTIONALLY INDEPENDENT OUTPUTS

Figure 3 shows the general structure of the proposed self-checking circuit.

The outputs $y_1, \ldots, y_m$ of the circuit f_c are arranged in h different groups $Y_1, \ldots, Y_h$ with $Y_i = \{y_{i_1}, \ldots, y_{i_{ni}}\}$, $i = 1, \ldots, h$, of independent outputs and unidirectionally independent outputs. Every group $Y_i = \{y_{i_1}, \ldots, y_{i_{ni}}\}$, $i = 1, \ldots, h$, will be supplemented with some additional outputs $v_{i_1}, \ldots, v_{i_{ti}}$ forming an extended group $Y_i = \{y_{i_1}, \ldots, y_{i_{ni}}, v_{h_1}, \ldots, v_{h_{ti}}\}$.

If the outputs of a group Y_j are independent only a single parity bit v_{j1} is added. In Fig. 3 Y_1 is assumed to be a group of independent outputs. Therefore only a single parity bit is added to this group.

If the outputs of a group Y_i are unidirectionally independent the t_i additional outputs are determined in such a way that the outputs $y_{i_1}, \ldots, y_{i_{ni}}$, $v_{h_1}, \ldots, v_{h_{ti}}$ are elements of an error detecting code for all unidirectional errors.

In Fig. 3 the group Y_h is supposed to be a group of unidirectionally independent outputs. Thus the outputs $v_{h_1}, \ldots, v_{h_{ti}}$ are added.

A possible error detecting code is a Berger-code[13] or an s-out-of-r code[14] where s is the number of 1's and r is the number of bits, $r = n_i + t_i$.

To determine the elements of an s-out-of-r code we consider all the possible binary n_i-tupels of the

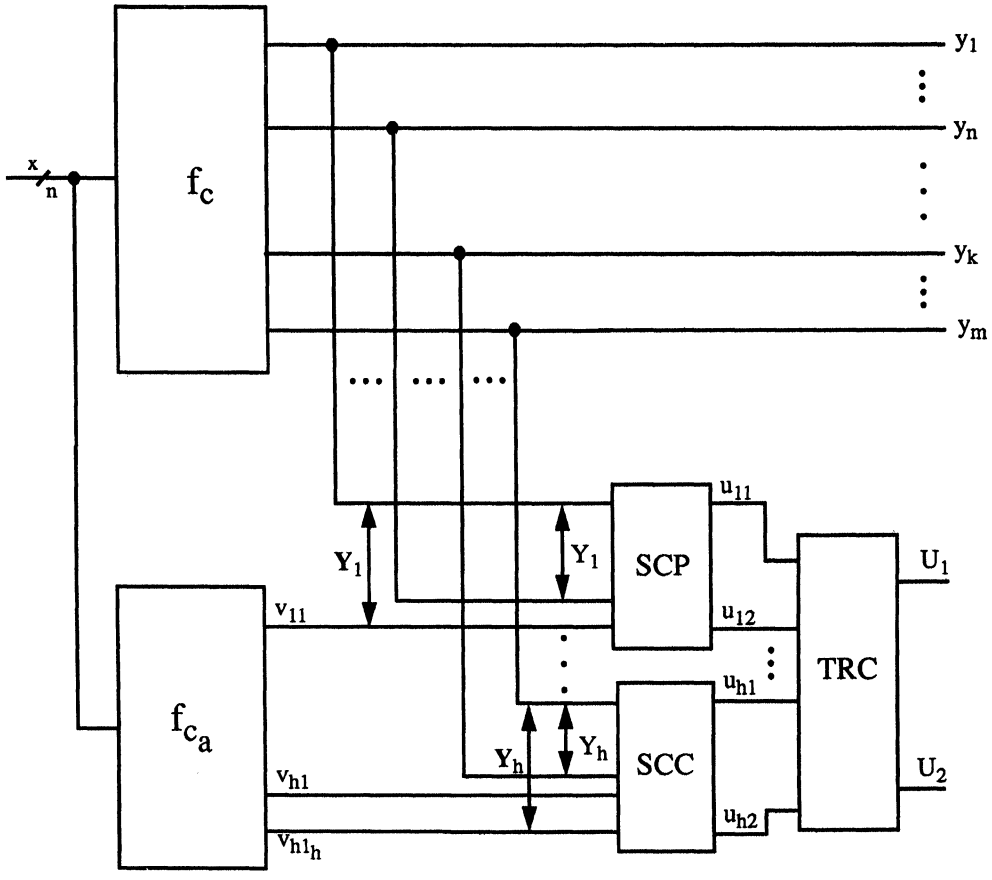

FIGURE 3 General structure of the proposed self-checking circuit.

outputs $y_{i_1}, \ldots, y_{i_{n_i}}$ of the group Y_i for $x \in \mathbf{X}$. Let M_i (m_i) be the maximal (minimal) number of 1's of this n-tupels. Then t_i is determined by $t_i = M_i - m_i$. It is easy to determine the outputs $v_{h_1} \ldots v_{h_{t_i}}$ such that $y_{i_1} \ldots y_{i_{n_i}}, v_{h_1} \ldots v_{h_{t_i}}$ are elements of an M_i-out-of-$(n_i + t_i)$ code.

If the number of 1's of the considered n_i-tuple $y_{i1}, \ldots, y_{ini}$ is K_i, $m_i \leq K_i \leq M_i$, then exactly $M_i - K_i$ additional outputs $v_{h1}, \ldots, v_{hti}$ have to be 1. These $M_i - K_i$ ones can be arbitrarily distributed over the additional outputs $v_{h1}, \ldots, v_{hti}$.

These additional outputs will be jointly implemented by use of an additional combinational circuit f_{ca} as shown in Fig. 3. Every fault in the additional circuit f_{ca} will be detected.

Every extended group $\mathbf{Y}_i$ is checked by use of a self-checking code-checker[2,3,15,16] with two outputs u_{i_1}, u_{i_2} with $u_{i_1} = \neg u_{i_2}$ in the fault free case. These outputs are compacted by a two-rail checker TRC with 2h inputs and 2 outputs U_1 and U_2, with $U_1 = \neg U_2$ in the fault free case. Every single gate fault of the circuit f_c and every fault of the circuit f_{ca} will be detected by an output $u_{i_1} = u_{i_2}$ for some i if it forces at least one of the outputs of f_c or f_{ca} to be erroneous for the first time. Since $u_{i_1} = u_{i_2}$ implies $U_1 = U_2$ at the output of the two-rail checker, every single gate fault of f_c and f_{ca} will be immediately detected. In Fig. 3 the group $\mathbf{Y}_1$ is checked by a self-checking parity checker SCP, and the group $\mathbf{Y}_h$ is checked by a self-checking code checker SCC for the corresponding code.

4. DETERMINATION OF INDEPENDENT AND UNIDIRECTIONALLY INDEPENDENT OUTPUTS

Pairs of independent outputs can be determined in the following steps:

1. Consider the nodes $n_1, \ldots, n_m$ of f_c. To every node n_j, $j = 1, \ldots, m$, we assign a set S_j. Thereby a node N_k, $N_k \in N$ belongs to S_j if there exists a path from N_k to n_j.
2. For every pair of nodes n_i, n_j with $i \neq j$, $i, j = 1, \ldots, n$, we determine the set $R_{i,j} = S_i \cap S_j$. If $R_{i,j} = \emptyset$ the outputs y_i and y_j are obviously independent. If $R_{i,j} \neq \emptyset$ we have to compute the expression $F'_{i,j;k}$ according to (1′). If for all k with $N_k \in R_{i,j}$, $R_{i,j} \neq \emptyset$ we have $F_{i',j;k} = 0$, the outputs y_i and y_j are independent.

For the determination of pairs of unidirectionally independent outputs this procedure has to be modified.

1′. To every node n_j, $j = 1, \ldots, m$, we assign two sets T_j and $T_j^{\neg}$. A node $N_k, N_k \in N$, belongs to T_j ($T_j^{\neg}$) if there exist a path from N_k to n_j with an even (odd) number of inversions.

2′. For every pair of nodes n_i, n_j with $i \neq j$, $i, j = 1, \ldots, n$, we determine the set $M_{i,j} = T_i \cap (T_j^{\neg}) \cup T_j \cap (T_i^{\neg})$. If $M_{i,j} = \emptyset$, the outputs y_i and y_j are obviously unidirectionally independent and the outputs y_i and y_j are called graphically unidirectionally independent. If $M_{i,j} \neq \emptyset$ we have to compute the expression $F_{i,j;k}$ according to (1).

If for all k with $N_k \in M_{i,j}$, $M_{i,j} \neq \emptyset$ we have $F_{i,j;k} = 0$, the outputs y_i and y_j are unidirectionally independent.

For the circuit f_c of Fig. 1 with the generalized circuit graph of Fig. 2, we have

$$S_1 = \{N_1, N_2, N_3, N_6, N_8\},$$
$$S_2 = \{N_2, N_3, N_6, N_7, N_9\},$$
$$S_3 = \{N_1, N_2, N_4, N_5, N_{10}\},$$
$$S_4 = \{N_2, N_5, N_6, N_7, N_{11}\},$$
$$S_5 = \{N_4, N_{12}\}$$

from which we conclude that

$$R_{1,5} = S_1 \cap S_5 = \emptyset, R_{2,5} = S_2 \cap S_5 = \emptyset,$$
$$R_{4,5} = S_4 \cap S_5 = \emptyset,$$

and the pairs of outputs y_1 and y_5, y_2 and y_5 and y_4 and y_5 are graphically independent.

For $j = 1, \ldots, 5$ the sets

$$T_j(T_j^{\neg})$$

are

$$T_1 = \{N_1, N_2, N_3, N_6, N_8\}, T_1^{\neg} = \emptyset,$$
$$T_2 = \{N_2, N_3, N_6, N_7, N_9\}, T_2^{\neg} = \{N_3\}$$
$$T_3 = \{N_1, N_2, N_4, N_5, N_{10}\}, T_3^{\neg} = \emptyset$$
$$T_4 = \{N_2, N_5, N_6, N_7, N_{11}\}, T_4^{\neg} = \emptyset$$
$$T_5 = \{N_{12}\}, T_5^{\neg} = \{N_4\}.$$

For

$$M_{i,j}, i \neq j, i, j = 1, \ldots, 5$$

we have

$$M_{12} = \{N_3\}, M_{13} = \emptyset, M_{14} = \emptyset, M_{15} = \emptyset, M_{23} = \emptyset,$$
$$M_{24} = \emptyset, M_{25} = \emptyset, M_{34} = \emptyset, M_{35} = \{N_4\}, M_{45} = \emptyset$$

and the pairs of output nodes

$$(n_1, n_3), (n_1, n_4), (n_1, n_5), (n_2, n_3), (n_2, n_4), (n_2, n_5),$$
$$(n_3, n_4) \text{ and } (n_4, n_5)$$

are graphically unidirectionally independent.

If we would like to determine not only the graphically unidirectionally independent pairs of outputs, the cases $M_{i,j} \neq \emptyset$, i.e. $M_{1,2} = \{N_3\}$ and $M_{3,5} = \{N_4\}$ have to be considered in more detail.

Since we have

$$F_{1,2;3} = \frac{dy_1}{dz_3}(x) \cdot \frac{dy_2}{dz_3}(x)(y_1(x) \oplus y_2(x)) = x_3 x_4 \neq 0$$

and

$$F_{3,5;4} = \frac{dy_3}{dz_4}(x) \cdot \frac{dy_5}{dz_4}(x)(y_3(x) \oplus y_5(x)) = 0,$$

the outputs y_3 and y_5 are unidirectionally independent and the outputs y_1 and y_2 are not unidirectionally independent.

Now we describe how (maximal) sets of independent outputs can be determined. To this end we draw a first dependency graph $\underline{G}^1 = (\underline{N}^1, \underline{V}^1)$. The nodes $\underline{N}^1$ of $\underline{G}^1$ are the outputs $y_1, \ldots, y_n$ of the considered circuit f_c. Two nodes $y_i, y_j \in \underline{N}^1$ are connected by an edge if the pair of nodes is (graphically) independent. Every group of independent outputs corresponds to a complete subgraph of $\underline{G}^1$, i.e. a clique of $\underline{G}^1$. The determination of a complete subgraph is a standard problem of graph theory.

In a similar way, we determine sets of unidirectionally independent outputs. We draw a second dependency graph $\underline{G}^2 = (\underline{N}^2, \underline{V}^2)$. The nodes $\underline{N}^2$ of $\underline{G}^2$ are again the outputs $y_1, \ldots, y_n$ of the considered circuit f_c. Two nodes $y_i, y_j \in \underline{N}^2$ are connected by an edge if these nodes are (graphically) unidirectionally independent. Every group of unidirectionally independent outputs corresponds to a complete subgraph of $\underline{G}^2$, i.e. a clique of $\underline{G}^2$. In the following example we restrict ourself to the graphically independent and graphically unidirectionally independent outputs.

The graph $\underline{G}^1$ of graphically independent pairs of outputs of f_c is given in Fig. 4. The three possible maximal complete subgraphs consist of the nodes n_1 and n_5, n_2 and n_5 and n_4 and n_5. Fig. 5 shows the graph $\underline{G}^2$ of the graphically unidirectionally

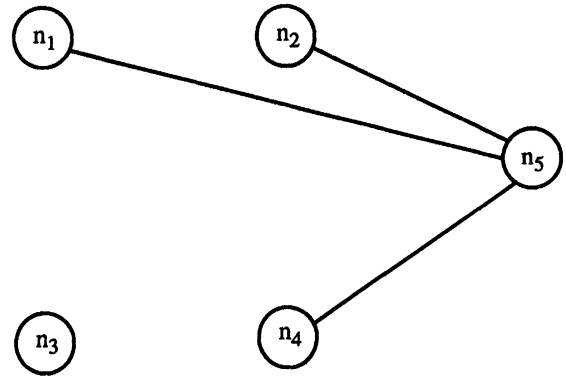

FIGURE 4 Graph of independent outputs.

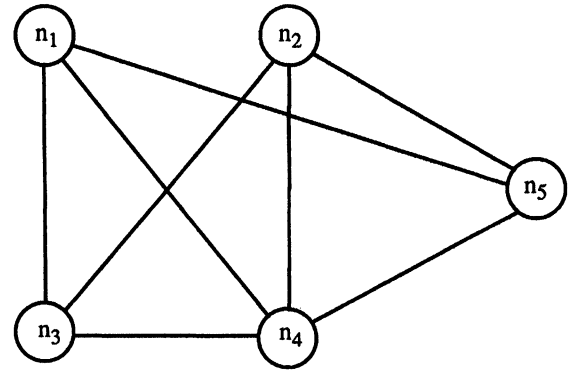

FIGURE 5 Graph of unidirectionally independent outputs.

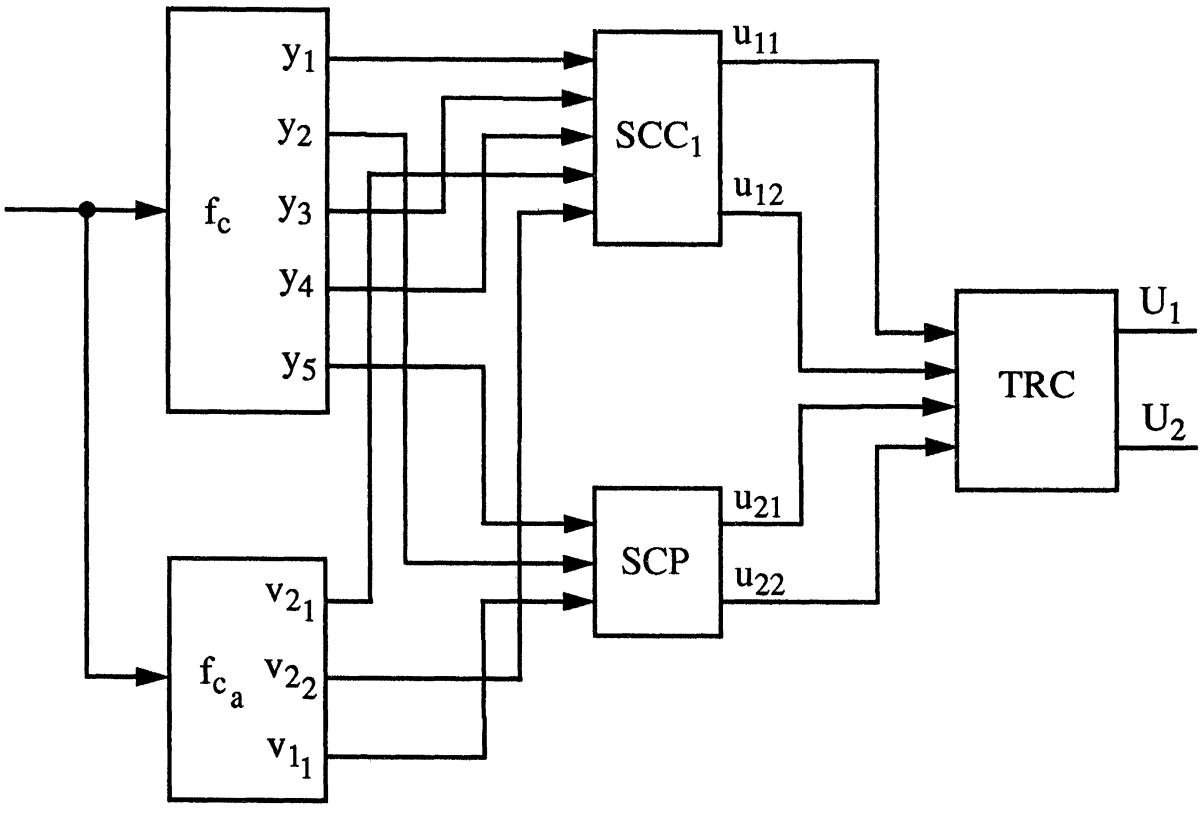

FIGURE 6 Self-checking circuit of the example of Fig. 1.

independent outputs of f_c. The maximal complete subgraphs of $\underline{G}^2$ consist of the sets of nodes $\{n_1, n_4, n_5\}$, $\{n_1, n_3, n_4\}$, $\{n_2, n_3, n_4\}$, $\{n_3, n_4, n_5\}$ and $\{n_2, n_4, n_5\}$.

We choose the set $Y_1 = \{n_2, n_5\}$ of graphically independent outputs and the set $Y_2 = \{n_1, n_3, n_4\}$ of graphically unidirectionally independent outputs. To the set $Y_1 = \{y_2, y_5\}$ of graphically independent outputs a single parity bit v_{1_1}, $v_{1_1} = \neg(y_2 \oplus y_5)$ is added. Thus we have $Y_1 = \{y_2, y_5, v_{1_1}\}$.

To the outputs y_1, y_3, y_4 of the group $\mathbf{Y}_2$ two additional outputs v_{2_1},v_{2_2} are added to form a Beger code with three information bits and two control bits. All the additional outputs v_{1_1},v_{2_1} and v_{2_2} can be jointly implemented. The outputs y_2, y_5, and v_{1_1} of the group $\mathbf{Y}_1$ are checked by a self-checking parity checker SCP. The group $\mathbf{Y}_2$ is checked by a self-checking Berger code checker SCC_1. The outputs of the SCC_1 and the SCP are compacted by a self-checking two-rail-checker TRC as shown in Fig. 6.

The resulting circuit is self-checking.

5. EXPERIMENTAL RESULTS

Table 1 showes the experimental results for the MCNC benchmarks.

The following abbreviations are used:

- inp: number of inputs
- out: number of outputs
- pgio: number of pairs of graphically independent outputs
- pguo: number of pairs of graphically unidirectionally independent outputs
- pio: number of pairs of independent outputs

TABLE I Experimental results

circuit	inp	out	pgio	pguo	pio	puo	go	bB	bP	bT	ao
1	2	3	4	5	6	7	8	9	10	11	12
cm138a	6	8	—	28	—	28	1	4	—	4	15
cu	14	11	33	54	33	54	2	5	—	5	75
cm162a	14	5	—	3	—	4	4	5	—	5	107
decod	5	16	27	120	27	120	1	5	—	5	<1
ldd	9	19	29	79	61	171	1	5	—	5	31
apex7	49	37	361	522	379	555	12	3	10	13	119
z4ml	7	4	—	—	3	6	3	—	3	3	62
x2	10	7	3	14	4	15	4	2	3	5	85
c8	28	18	96	152	96	152	2	6	—	6	156
f51m	8	8	1	5	1	7	4	6	1	7	55
ttt2	24	21	105	116	128	148	7	4	5	9	79
vda	17	39	11	17	169	343	14	26	—	26	60
pm1	16	13	63	68	69	74	2	3	1	4	88

puo: number of pairs of unidirectionally independent outputs
go: number of groups of independent and unidirectionally independent outputs
bB: number of additional bits for Berger Codes
bP: number of additional parity bits
bT: total number of additional bits (bT = bB + bP)
ao: additional area overhead in percent of the area of the original circuit

The pairs of graphically independent outputs pgio (column 4) and the pairs of graphically unidirectionally independent outputs pguo (column 5) are determined by use of the modified generalized circuit graph. The pairs of independent outputs pio (column 6) and the pairs of unidirectionally independent outputs puo (column 7) are determined according to (1) and (1′) respectively where the corresponding Boolean derivations have to be taken into account.

The Boolean derivations in (1) and (1′) are computed by use of the program system MIS[17]. The corresponding Boolean functions are represented as BDD's. Graphically (unidirectionally) independent outputs are always (unidirectionally) independent. Therefore, the number of pairs of (unidirectionally) independent outputs is always greater than or equal to the number of pairs of graphically (unidirectionally) independent outputs.

The minimal number of mutually disjoint groups of independent and unidirectionally independent outputs go are given in column 8.

Groups of unidirectionally independent outputs are monitored by Berger Codes. The number of necessary additional bits bB is represented in column 9. Groups of independent outputs are checked by additional parity bits. The number of additional parity bits bP is given in column 10. The total number of check bits bT (Berger code bits and parity bits) is given in column 11 of Table I. The area overhead of the additional circuit f_{c_a} is shown in the row 12 of Table. The additional circuit f_{c_a} was sythesized by use of the structurally given circuit f_c in the following way: If the outputs of a group $Y_i = \{y_{i_1}, \ldots, y_{i_{hi}}\}$ are independent a single additional output v_{i_1} of the parity bit is determined by connecting the outputs $y_{i_1}, \ldots, y_{i_{hi}}$ by XOR-gates. If the outputs of a group $Y_j = \{y_{j_1}, \ldots, y_{j_{hj}}\}$ are unidirectionally independent the outputs $\{y_{j_1}, \ldots, y_{j_{hj}}\}$ are connected to a counter of zeros to determine the additional bits of a Berger code. Then all the outputs of the circuit f_{c_a} are jointly optimized by use of the standard MIS-algorithm script.rugget.

The necessary computing time on a SUN Workstation Spark 10 including the determination of pairs of independent and unidirectionally independent outputs, the determination of maximal complete subgraphs and the synthesis of the additional

circuit was in the range of 0.33 to 3376.46 seconds. The average additional area overhead is 71%.

6. CONCLUSIONS

In this paper a new structure dependent method for the design of self-checking and self-testing combinational circuits was proposed. Maximal groups of independent outputs and unidirectionally independent outputs are determined. It is shown how pairs of (unidirectionally) independent outputs can heuristically be determined by a simple graph–theoretical algorithm. Maximal groups of (unidirectionally) independent outputs can be computed as cliques of a dependency graph, which is a standard graph–theoretical problem. A single parity bit is added to every group of independent outputs. A few additional outputs are added to every group of unidirectionally independent outputs in such a way that these groups of unidirectionally independent outputs and the corresponding additional outputs are elements of a unidirectional error detecting code. This method is explained in detail for a simple example. The applicability of the proposed method is demonstrated for the MCNC benchmarks.

Acknowledgment

The authors are very grateful to the reviewers of this paper for their helpful comments.

References

[1] P. K. Lala, Fault Tolerant and Fault Testable Hardware Design. Prentice Hall, Englewood-Cliffs, N.J.

[2] V. V. Sapozhnikov and Vl. V. Sapozhnikov, Self-Checking Checkers for Balanced Codes. *Automation and Remote Control*, vol. 53, No 3, part 1, 321–348 (1992).

[3] V. V. Saposhnikov and Vl. V. Saposhnikov, Self-Checking Discrete Circuits (in russ.) Energoatomisdat, St. Petersburg (1992).

[4] S. K. Gupta and D. K. Pradhan, Can Concuent Checkers Help BIST?, *Proc. 1992 Intern. Test Conference*, pp. 140–150 (1992).

[5] R. M. Sedmak, Design for Self-Verification. An Approach for Dealing with Testability Problems in VLSI-based Design. *Proc. 1979 IEEE Test Conference*, pp. 112–120 (1979).

[6] E. Fujiwara, N. Muto and K. Matsuoka, A Self-Testing Group Parity Prediction Checker and its Use for Built-in-Testing. *IEEE Trans. Comp.*, vol. C-33, No 6, 578–583 (1984).

[7] T. R. N. Rao and E. Fujiwara, Error Control Coding for Computer Systems, Prentice Hall (1989).

[8] E. S. Sogomonyan, Reliability of Self-Testing using Functional Diagnosic Tools. *Automation and Remote Control*, vol. 49, No 10, Part 2, October 1988, pp. 1376–1380.

[9] E. S. Sogomonyan, Design of Built-In Self-Checking Monitoring Circuits for Combinational Devices. *Automation and Remote Control*, vol. 35, No 2, part 2, 280–289 (1974).

[10] E. Fujiwara, A Self-Testing Group Parity Prediction Checker and its Use for Built-in Testing. *Proc. 13th Symposium Fault-Tolerant Computing*, Milano, pp. 146–153 (1983).

[11] E. S. Sogomonyan and M. Goessel, Design of Self-Testing and On-line Fault Detection Combinational Circuits with Weakly Independent Outputs. *JETTA*, **4**, 267–281 (1993).

[12] F. Y. Busaba and P. K. Lala, Self-Checking Combinational Circuit Design for Single and Unidirectional Multibit Errors, *JETTA*, **5**, 19–28 (1994).

[13] J. M. Berger, A Note on Error Detecting Codes for Asymmetric Channels. *Information and Control*, **4**, 68–73 (1991).

[14] D. A. Anderson and G. Metze, Design of Totally Self-Checking Check Circuits for m-out-of-n Codes. *IEEE Trans. Comp.*, vol. C-22, 263–269 (1973).

[15] M. A. Marouf and A. D. Friedman, Design of Self-Checking Checkers for Berger-Codes. *Proc. Int. Symp. Fault-Tolerant Computing*, 179–184 (1978).

[16] S. Kundu and S. M. Reddy, Embedded Totally Self-Checking Checkers. A Practical Design. *IEEE Design and Test of Computers*, August 1990, 5–16.

[17] R. K. Brayton, R. Rudell, A. Sangiovanni-Vincentelli and A. R. Wang, "MIS: A multi-level logic optimization program," *IEEE Trans. Computer-Aided Design*, **7**, 1062–1081.

Authors' Biographies

Andrej Morosow received the M.S. degree in electrical enigneering from the University of Railway Engineering, St. Petersburg, Russia, in 1992. From 1992 to 1996 he was with Max-Planck-Society Fault Tolerant Computing Group at the University Potsdam, Germany. Now he is with the Computer Science Department of the University of Potsdam, Germany, where he is working toward his Ph.D. degree. His research interest are the design of self-checking and self-testing circuits.

Prof. Valerij Saposhnikov received the M.S. degree (1963), the Ph.D. degree (1968) and the doctoral degree (1982) all in electrical engineering from the University of Railway Engineering,

St. Petersburg (formerly Leningrad), Russia. In 1982 he became a professor, and in 1989 a vice president of this university. He published as an author and coauthor about 250 scientific papers and 8 books in russian language in the field of synthesis, diagnosis, self-checking circuits and control systems for railway transportation. Since 1992 he is a visiting professor of the Max-Planck Fault Tolerant Computing Group at the University of Potsdam, Germany. He is a member of the Russian Transportation Academy and of the International Academy of Higher Education.

Prof. Vladimir Saposhnikov received his M.S. degree (1963), the Ph.D. degree (1969) and the doctoral degree (1984) all in electrical engineering from the University of Railway Engineering St. Petersburg (formerly Leningrad), Russia. Since 1987 he is a professor and since 1991 the head of a department. Since 1992 he is a visiting professor of the Max-Planck Fault Tolerant Computing Group at the University of Potsdam, Germany. He is the author and coauthor of about 300 scientific publications and 8 books in Russian language. His research interests are synthesis and diagnosis of discrete systems, self-checking systems and fail-safe transportation systems. He is a member of the Russian Transportation Academy and of the International Academy of Higher Education.

Prof. Michael Goessel received the M.S. Degree (1965) and the Ph.D. Degree (1968), both in physics, from the University of Jena, Germany and the doctoral degree in Computer Science (1971) from the Technical University of Dresden, Germany. From 1966 to 1968 he was a research assistent at the University of Leningrad. From 1969 to 1991 he was with the Central Institute of Cybernetics (ZKI) of the Academy of Sciences, Berlin, of the formerly GDR. In 1992 he was appointed as the head of the Fault Tolerant Computing Group of the Max-Planck-Society at the University of Potsdam, Germany. Since 1994 he is a full-time professor of Computer Science at the University of Potsdam, Germany. He is the author and coauthor of 7 books, including "Error Detection Circuits", (Goessel, Graf, McGraw-Hill 1993) and "Memory Architecture and Parallel Access", (Goessel, Rebel, Creutzburg, Wiley 1994).