VLSI DESIGN
2000, Vol. 11, No. 1, pp. 1–21
Reprints available directly from the publisher
Photocopying permitted by license only

New Self-dual Circuits for Error Detection and Testing

A. DMITRIEV[a], V. SAPOSHNIKOV[b], VL. SAPOSHNIKOV[b], M. GOESSEL[a], VL. MOSHANIN[a] and A. MOROSOV[a,*]

[a]*University of Potsdam, Department of Computer Science, Fault Tolerant Computing Group, PSF 601553, D-14415 Potsdam, Germany;* [b]*Railway Transportation State University, Moskovskij pr. 9, SU190031 St.-Petersburg, Russia*

(Received 1 April 1999; In final form 5 October 1999)

In this paper new methods for the transformation of a given combinational circuit into a self-dual circuit based on the notion of a self-dual complement are investigated. The large variety of self-dual complements can be utilized to optimize the transformed self-dual circuit. Self-dual duplication and self-dual parity prediction are considered in detail. As a method for the reduction of self-dual outputs, output space compaction of self-dual outputs is considered. For the first time we also describe in this paper how a self-dual circuit can be modified into a self-dual fault-secure circuit.

Keywords: Error detection, testing, on-line testing, self-dual circuits, output space compaction, fault-secure circuits

INTRODUCTION

Self-checking and self-testing circuits are becoming more and more important as the complexity of VLSI continues to increase. For the design of self-checking or self-testing circuits very often coding techniques as described, for example in [1–8] with additional control bits and additional checkers are used. On the other hand, time redundancy is also utilized for error detection.

Especially for arithmetic operations with simple functional equations the method of pseudo-duplication is of interest [4, 9–11]. The data are processed twice in succession but along different data paths, latched and compared.

The method of alternating inputs, introduced in [12] and further developed in [13] may be considered as an interesting general method for the combination of time- and hardware- redundancy also for random logic. According to [12, 13] a given combinational circuit is transformed into a self-dual circuit. For a large percentage of the considered benchmark circuits the necessary area overhead is very small. Since always the corresponding alternating inputs x (original input vector) and $\bar{x}$ (inverted input vector) are successively submitted to

*Corresponding author. Fax: +49-331-9771190.

the self-dual circuit time redundancy is 100% and the method is mainly applicable for the design of control systems for which time is not a critical issue. A great advantage of the method of alternating inputs is that stuck-at faults at the input lines of the monitored circuit are also detected by this method.

Self-dual circuits can be used in different modes of operation.

In on-line mode the original functional inputs x and their inverted values $\bar{x}$ are subsequently submitted to the self-dual circuit. For these alternating inputs the corresponding outputs are alternating as long as no error occurs.

In test mode it is not necessary to store the test outputs of the circuit under test. The elements x_t of the test set T and their inverted values $\bar{x}_t$ are applied to the self-dual circuit. A fault is detected if the corresponding outputs are not alternating.

In fast mode, without error detection, only the functional inputs are applied to the circuit.

Thus, the application of self-dual circuits and the combination of their different modes of operation allows one to achieve different levels of error detection and fault tolerance.

In this paper new methods for the transformation of a given combinational circuit into a self-dual circuit are investigated. These methods are based on the concept of a self-dual complement of a Boolean function. Thereby a Boolean function $\delta_f(x)$ is called a self-dual complement of the Boolean function $f(x)$ if the modulo-2-sum $f(x) \oplus \delta_f(x)$ of $f(x)$ and $\delta_f(x)$ is self-dual. To modify a given combinational circuit f_c into a self-dual circuit f_{sd} the original circuit is completed by a circuit $\delta_{fc}(x)$ implementing the self-dual complement $\delta_f(x)$. For $x = x_1, \ldots, x_m$ there exist $2^{2^{m-1}}$ different self-dual m-ary Boolean functions. Thus for a given Boolean function $f(x)$ there also exist $2^{2^{m-1}}$ different self-dual complements. This variety of self-dual complements can be utilized to optimize the modified self-dual circuit. An optimization of the self-dual circuit with such a variety of choices is possible even if the original circuit is completely specified and no don't-care conditions of the original circuit are available. The original circuit and the carefully chosen self-dual complement can be jointly or separately implemented and optimized.

First, self-dual duplication is described. To reduce the number of self-dual outputs, self-dual parity prediction is investigated. As a further method for the reduction of self-dual outputs, output space compaction of self-dual outputs is considered.

A fault ϕ of a self-dual circuit is detected, if, for alternating inputs the outputs of the erroneous circuit are not alternating. Faults of the primary input lines are also detected in test mode and in on-line mode with a high probability. A fault will not be detected if the outputs are erroneous but alternating. For safety-critical applications such a situation can not be tolerated. For the first time, we also describe in this paper how a self-dual circuit can be modified into a self-dual fault-secure circuit, *i.e.*, into a self-dual circuit for which every single gate fault will immediately be detected by not being alternating if the fault, for the first time, forces a circuit output to be erroneous.

In general, it is demonstrated in this paper that the method of modifying a given combinational circuit into a self-dual circuit by use of a self-dual complement is an effective method for error detection and testing with low area overhead and a high fault coverage. The paper is organized as follows.

The next chapter contains the basic notions and notations which are applied in the following chapters for the design of different types of self-dual circuits. First, the notion of a self-dual complement of a Boolean function is introduced and self-dual complements with both a minimal number of ones and a maximal number of ones are described. These types of self-dual complements are especially useful for the design of self-dual circuits. Self-dual testability is formally defined and it is shown how a self-dual complement can be used for error detection. Self-dual fault secureness is defined both at a functional and at a structural level.

In third chapter self-dual duplication is investigated. For every circuit output of the original circuit a self-dual complement is determined. Separate and joint implementations of the self-dual complements and the functional circuit are considered in detail. Experimental results for the area overhead and fault coverage in test mode and for the probability of not detecting an error in on-line mode are obtained for benchmark circuits.

The fourth chapter deals with self-dual parity. Only the ordinary parity prediction function is replaced by a self-dual complement of this function. As in the third chapter, the separate implementations of the self-dual complement and the functional circuit are compared with the corresponding joint implementation of these functions. The experimental results compare area overhead, fault coverage in test mode, and error detection probability in on-line mode for ordinary parity prediction with the results obtained for self-dual parity.

Linear output space compaction of alternating signals is described briefly in the fifth chapter.

The transformation of an arbitrary self-dual circuit into a fault-secure self-dual circuit is presented in the sixth chapter.

The last chapter contains conclusions.

BASIC NOTIONS AND NOTATIONS

In this chapter, we introduce the notion of a self-dual complement of a Boolean function and we describe basic properties of self-dual complements which are useful for error detection and testing.

Let g: $X \rightarrow Y$, $X=\{0,1\}^m, x=x_1,\ldots,x_m$ be an m-ary Boolean function. Then $g_d(x)=\overline{g}(\overline{x})$, $\overline{x}=\overline{x}_1,\ldots,\overline{x}_m$ is called the dual function of g. The Boolean function g is called self-dual if $g(x)=g_d(x)=\overline{g}(\overline{x})$ for $x \in X$. Till now the only method for the transformation of a given combinational circuit into a self-dual circuit is based on the fact [12], that an arbitrary Boolean function $g(x)$ can easily be transformed into a self-dual Boolean function $G(a,x)$

$$G(a,x)=\overline{a}g(x) \vee a\overline{g}(\overline{x}) \tag{1}$$

with $G(0,x)=g(x)$ and $G(1,\overline{x})=\overline{g}(x)$. Thereby $a \in \{0,1\}$ is an additional Boolean variable. In [12] the original Boolean function $g(x)$ and its dual function $g_d(x)=\overline{g}(\overline{x})$ are combined to form the self-dual Boolean function $G(a,x)$. In [13] it was shown how this method can be applied to the transformation of a combinational circuit which is given as a netlist of AND- and OR-gates and INVERTERS. If in the original circuit all the AND-gates are replaced by OR-gates and if all the OR-gates of the original circuit are replaced by AND-gates the modified (dual) circuit implements the corresponding dual function of the original circuit at its outputs. Now the outputs of the original circuit and the corresponding outputs of the modified dual circuit are connected as it shown in Figure 1, and the resulting circuitry implements at its outputs the desired self-dual functions.

To reduce the necessary hardware overhead the original circuit and the modified dual circuit can be jointly implemented and optimized. The resulting average area overhead is 70% of the original circuit [13].

A Boolean function can also be transformed into a self-dual function by use of a self-dual complement which is introduced now. By use of this method the necessary area overhead can be reduced.

DEFINITION 1 Let g be a Boolean function. Then δ_f is said to be a self-dual complement of g if $h=g \oplus \delta_f$ is self-dual.

It is easy to prove that for $i=1,\ldots,m$ the Boolean function

$$\delta_i(x)=x_i(g(x) \oplus \overline{g}(\overline{x})) \tag{2}$$

is a self-dual complement of $g(x)$. x_i is one of the m components of the input vector $x=(x_1,\ldots,x_m)$ of the m-ary Boolean function $g(x)$. For these functions $\delta_i(x)=x_i(g(x) \oplus \overline{g}(\overline{x}))$, $i=1,\ldots,m$ the

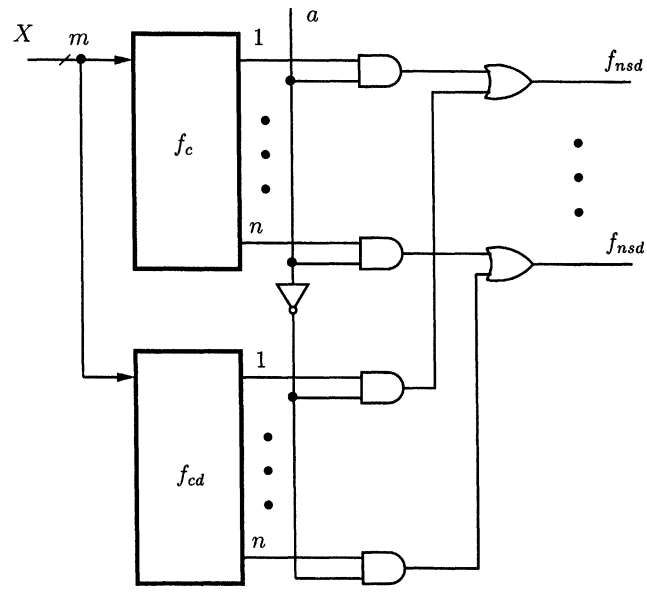

FIGURE 1 Connection of the original and modified circuits before circuit optimization.

number of values x for which $\delta_i(x)=1$ is minimal [14].

Another interesting class of self-dual complements of the function $g(x)$ are the functions

$$\tilde{\delta}_i(x) = x_i \vee (g(x) \oplus g(\bar{x})), \quad i = 1, \ldots, m, \quad (3)$$

for which the number of values with $\tilde{\delta}_i(x) = 1$ is maximal [14].

Now we describe how a self-dual complement of a Boolean function can be utilized for error detection. 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. In this paper we suppose that a technical fault $\phi \in \Phi$ can always be expressed at a gate level in such a way that in the presence of the fault ϕ a fault free gate g_q implementing the Boolean function $G_q(x)$ is transformed into a faulty gate $g_{q\phi}$ implementing the erroneous Boolean function $G_q(\phi, x)$. We are only interested in unidirectional gate faults which are now described.

A fault ϕ is called a unidirectional gate fault of gate g_q if either the correct output 0 for some inputs $x \in \check{X} \subseteq X$ is changed into an erroneous output 1 and the correct output 1 remains correct or the correct output 1 for some inputs $x \in \hat{X} \subseteq X$ is changed into an erroneous output 0 and the correct output 0 remains correct. For an unidirectional gate fault only one of the two possible output values of a faulty gate may be erroneous. For single and multiple stuck-at-0/1 faults at the input- and output-lines these conditions are always satisfied for AND-, OR-, NAND- and NOR-gates, and INVERTERS.

A bridging fault between lines of different gates, an intermittent value of a CMOS-gate, the replacement of an AND-gate by a NAND-gate, or a stuck-at fault at an input line of an XOR-gate if, because of a fanout, this fault is not equivalent to a stuck-at fault at the output of the preceding gate, are examples of faults which are not unidirectional gate faults.

If a fault $\phi_i \in \Phi$ is present, the output y_j of the circuit f_c under input x is denoted by $y_{j,i} = f_j(\phi_i, x)$. Thus, in the absence of a fault, we have the correct output $y_{j,0} = f_j(\phi_0, x) = f_j(x)$.

For error detection in a circuit implementing a Boolean function $g(x)$, a self-dual complement $\delta(x)$, is determined and g, δ, and $h(x) = g(x) \oplus \delta(x)$ are implemented as shown in Figure 2.

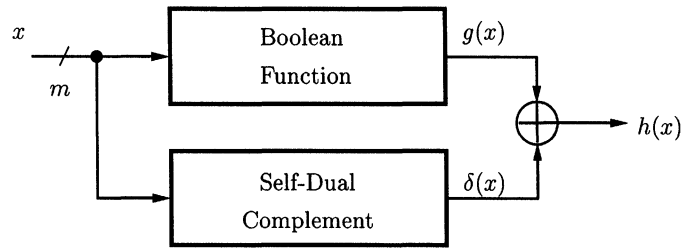

FIGURE 2 Error detection by use of self-dual complement.

We assume that the input $x = x_1, \ldots, x_m$ of the considered circuit is alternating. Since $h(x)$ is self-dual with $h(x) = \bar{h}(\bar{x})$ the output $y = h(x)$ is also alternating as long as no error occurs.

A fault is detected under the alternating inputs x and $\bar{x}$ if the outputs $h(x)$ and $h(\bar{x})$ are not alternating. The testability of a fault is now defined more precisely.

DEFINITION 2 A fault ϕ changing $g(x)$ into $g(\phi, x)$ is called self-dual testable if there exists an input x for which we have

$$g(\phi, x) \oplus \delta(x) = g(\phi, \bar{x}) \oplus \delta(\bar{x}).$$

Thus in the case of a testable fault for the alternating inputs x and $\bar{x}$, the output y is not alternating. A fault ϕ changing $g(x)$ into $g(\phi, x)$ is self-dual testable if there exists an input x' such that we have $g(x') \neq g(\phi, x')$ and $g(\bar{x}') = g(\phi, \bar{x}')$. A fault ϕ can not be detected under the alternating inputs x and $\bar{x}$ if the outputs of the erroneous circuit are alternating. This situation occurs if either the alternating outputs of the erroneous circuit are both correct or both erroneous. In the later case erroneous circuit outputs are not detected as erroneous by alternating inputs. Especially in on-line mode for safety-critical applications such a situation should be avoided. If every error due to a technical fault $\phi \in \Phi$ is immediately detected the circuit is called fault-secure.

Now let us consider a self-dual combinational circuit f_c implementing at its outputs $y_1, \ldots, y_n$ the self-dual Boolean functions $y_1 = f_1(x), \ldots, y_n = f_n(x)$.

Then the output y_j, $j \in \{1, \ldots, n\}$, is self-dual fault-secure with respect to the fault $\phi \in \Phi$ if, in the presence of this fault, for every $x \in X$ the output y_j is never simultaneously erroneous under the alternating inputs x and $\bar{x}$. If, due to the fault ϕ, an output signal is erroneous the signal is not alternating.

We say: A self-dual circuit is completely fault-secure with respect to a fault $\phi \in \Phi$ if every output y_j, $j \in \{1, \ldots, n\}$ is a self-dual fault-secure with respect to the fault.

The self-dual circuit f_c is self-dual fault-secure with respect to the set Φ of faults if every output y_j, $j \in \{1, \ldots, n\}$ is self-dual fault-secure with respect to every fault $\phi \in \Phi$.

As an example let us consider the circuit of Figure 3 implementing the self-dual functions $y_1 = x_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1 x_2 \bar{x}_3 \vee \bar{x}_1\bar{x}_2 x_3 \vee x_1 x_2 x_3$, $y_2 = x_2 x_3 \vee x_2\bar{x}_3$.

The truth tables of the correct circuit (column A), the faulty circuit with a stuck-at-0 fault ϕ_1 at the output of gate 4 (column A_1), and the faulty circuit with a stuck-at-1 fault ϕ_2 at the output of gate 4 (column A_2) are given in Table I.

In the presence of the fault ϕ_1 the output y_1 is erroneous under the alternating pairs of inputs 010, 101 and 011, 100. Thus the output y_1 is not self-dual fault-secure with respect to the stuck-at-0 fault ϕ_1. On the other hand the output y_2 is erroneous under input 011 but not under input 100 and erroneous under input 101 but not under input 010. Therefore y_2 is self-dual fault-secure with respect to the stuck-at-0 fault ϕ_1. Both the outputs, y_1 and y_2 are self-dual fault-secure with respect to the single stuck-at-0 fault ϕ_2.

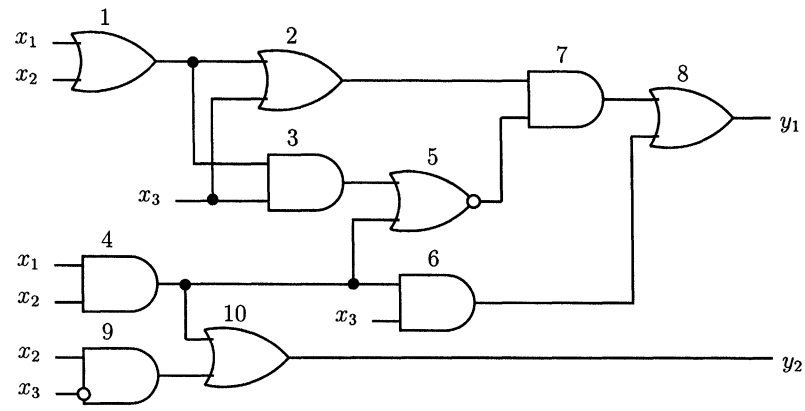

FIGURE 3 A self-dual circuit.

TABLE I Truth table of the circuit of Figure 3

			A		A_1		A_2	
x_1	x_2	x_3	y_1	y_2	y_1	y_2	y_1	y_2
0	0	0	0	0	0	$0\rightarrow 1$	0	0
0	0	1	1	0	1	$0\rightarrow 1$	1	0
0	1	0	1	1	$1\rightarrow 0$	1	1	0
0	1	1	0	1	$0\rightarrow 1$	1	0	$1\rightarrow 0$
1	0	0	1	0	$1\rightarrow 0$	$0\rightarrow 1$	1	0
1	0	1	0	0	$0\rightarrow 1$	$0\rightarrow 1$	0	0
1	1	0	0	1	0	1	$0\rightarrow 1$	1
1	1	1	1	1	1	1	$1\rightarrow 0$	$1\rightarrow 0$

In the following we are interested in the redesign of self-dual circuits into self-dual fault-secure circuit with respect to all single gate faults. Since gates are the elementary units of the design process we introduce the notion of a self-dual fault-secure gate.

DEFINITION 3 A gate g of a self-dual circuit f_c is called self-dual fault secure with respect to an output y_j of f_c if the output y_j of f_c is self-dual fault-secure with respect to all the considered single gate faults of g.

A gate g of a self-dual circuit f_c is called self-dual fault-secure if g is self-dual fault-secure with respect to every output y_j, $j \in 1, \ldots, n$, of f_c.

In Theorem 1, the necessary and sufficient conditions for a circuit output to be self-dual fault-secure with respect to a single gate fault are given.

THEOREM 1 *Let f_c be a self-dual combinational circuit. Let ϕ be a single gate fault of g_j of f_c changing the correct output $G_j(x)$ of g_j into the erroneous output $G_j(\phi, x)$ and let z_j denote the output of g_j. Then the circuit output y_j of f_c is (functionally) self-dual fault-secure with respect to ϕ if and only if we have*

$$\frac{dy_j}{dz_i}(x) \cdot \frac{dy_j}{dz_i}(\bar{x})[G_j(x) \oplus G_j(\phi, x)] \\ [G_j(\bar{x}) \oplus G_j(\phi, \bar{x})] = 0. \tag{4}$$

The proof is omitted here. Now we mention the following simple observation.

OBSERVATION 1 *A unidirectional gate fault cannot be detected at the output y_j of f_c under the alternating input pair x, $\bar{x}$ if the fault is simultaneously stimulated by x and $\bar{x}$ and if, from the*

location of the fault, two different paths to the output y_j of the circuit with a different parity of the number of inverters are sensitized by x and $\bar{x}$ respectively (see [12] for a preliminary formulation of this observation).

The transformation of a self-dual circuit into a self-dual fault-secure circuit relies very much on the following theorem.

THEOREM 2 *Let f_c be a self-dual combinational circuit with n binary outputs $y_1, \ldots, y_n$. If for every gate g of f_c the parity of the number of inverters of every path from g to the output y_j is equal then f_c is self-dual fault-secure with respect to the output y_j.*

Proof A unidirectional gate fault $G(\phi, x)$ can be propagated to a circuit output y_j under the alternating inputs x and $\bar{x}$ only along paths with the same parity of inverters. If only the input x (but not $\bar{x}$, or *vice versa*) stimulates the fault ϕ at most $y_j(\phi, x)$ (but not $y_j(\phi, \bar{x})$) is erroneous. Either the fault ϕ will be detected or both the output values will be correct.

Let us assume now that both x and $\bar{x}$ stimulate a unidirectional gate fault ϕ resulting, for example, in an erroneous value 1 instead of a correct 0. Such a situation is usually described [15] by use of the D-calculus as D. Then we have $G(\phi, x) = G(\phi, \bar{x}) = 1 \neq G(x) = G(\bar{x}) = 0$. Let us further assume that both x and $\bar{x}$ sensitized paths p_1 and p_2 from the location of the fault ϕ to the output y_j. These paths must have the same, for instance, an even parity of inverters. Then the stimulated erroneous D will be propagated under both the inputs x and $\bar{x}$ as an erroneous D to the output y_j changing both the correct values $y_j(x) = y_j(\bar{x}) = 0$ into the erroneous value $y_j(\phi, x) = y_j(\phi, \bar{x}) = 1$. This is a contradiction since f_c is self-dual with $y_j(x) \neq y_j(\bar{x})$.

A similar result was already obtained in [12].

■

According to Theorem 2, a unidirectional gate fault of a gate g can not be propagated to a circuit output y_j if all the paths from g to this circuit output have either an even or an odd number of inverters. If Theorem 2 is true for every output y_j, $j = 1, \ldots, n$, then the corresponding self-dual circuit is self-dual fault-secure.

By use the Theorem 2 the transformation of a self-dual circuit into a self-dual fault-secure circuit can be simplified if fault secureness is considered at a structural level only.

DEFINITION 4 A gate g of the circuit f_c is called structurally self-dual fault-secure with respect to a circuit output y_j if all the paths from gate g to the output y_j have either an even or an odd number of inverters.

A gate g of the circuit f_c is called completely structurally self-dual fault-secure if all the paths from the gate g to all the outputs y_j, $j = 1, \ldots, n$, have either an even or an odd number of inverters.

These notions, definitions and theorems will be applied in the following chapters for the design of self-dual circuits for error detection and testing.

SELF-DUAL DUPLICATION

In this section we describe the method of self-dual duplication [16]. To each of the functional circuit outputs of the original circuit a corresponding self-dual output is determined. The self-dual outputs are implemented by use of self-dual complements of the corresponding outputs of the original circuit. The self-dual complements and the monitored circuit itself can be separately or jointly implemented. A joint implementation and a careful selection of the appropriate self-dual complements of the circuit outputs significantly reduces the necessary area overhead for the proposed method of error detection. The proposed approach is different from "normal" duplication and comparison where both the original circuit and its duplicate have to be separately implemented.

The general principle of self-dual duplication is illustrated in Figure 4. The combinational circuit f_c implements the n-tupel $f(x) = f_1(x), \ldots, f_n(x)$ of m-ary Boolean functions $f_1, \ldots, f_n$ and f_c has the outputs $y_1, \ldots, y_n$. For every output y_j, $j = 1, \ldots, n$

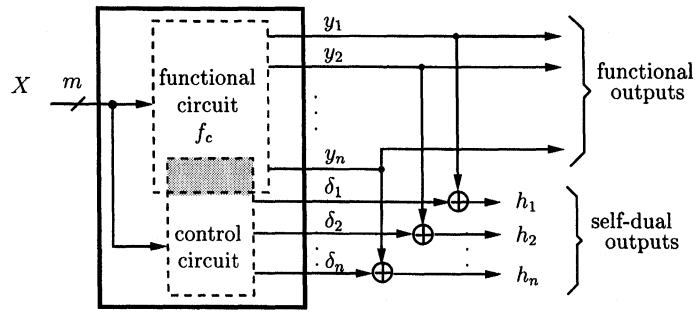

FIGURE 4 General principle of self-dual duplication.

of f_c a self-dual complement δ_j according to Definition 1 is determined.

For $j=1,\ldots,n$ the Boolean function $h_j(x)=f_j(x)\oplus\delta_j(x)$ is self-dual. The original circuit and the self-dual complements are separately or jointly implemented.

As an example we consider the self-dual duplication of a circuit f_c with four inputs and three outputs implementing the Boolean functions:

$$\begin{aligned} y_1 &= f_1(x_1,x_2,x_3,x_4)\\ &= \bar{x}_1x_2x_3\bar{x}_4 \vee (x_1 \vee x_2)x_3x_4 \vee \bar{x}_1\bar{x}_3x_4\\ y_2 &= f_2(x_1,x_2,x_3,x_4)\\ &= x_3\bar{x}_4 \vee x_1x_2(x_3 \vee \bar{x}_4) \vee \bar{x}_1\bar{x}_2(x_3 \vee \bar{x}_4)\\ y_3 &= f_3(x_1,x_2,x_3,x_4)\\ &= \bar{x}_1\bar{x}_2x_3 \vee \bar{x}_1x_2x_4 \vee x_1x_2\bar{x}_3 \vee x_1\bar{x}_2x_4 \end{aligned}$$

The original circuit is shown in Figure 5.

The self-dual complements of the outputs y_1, y_2, and y_3, are determined according to (2) as:

$$\begin{aligned} \delta_1(x_1,x_2,x_3,x_4) &= x_1(f_1(x)\oplus\bar{f}_1(\bar{x})) = x_1x_2\bar{x}_3\\ \delta_2(x_1,x_2,x_3,x_4) &= x_3(f_2(x)\oplus\bar{f}_2(\bar{x})) = x_3x_4\\ \delta_3(x_1,x_2,x_3,x_4) &= x_1(f_3(x)\oplus\bar{f}_3(\bar{x})) = x_1x_2 \end{aligned}$$

A joint implementation of the original circuit and the self-dual complements is represented in Figure 6. The circuits in Figures 5 and 6 are optimized by the tool SIS *script.rugged* [17]. The circuit of Figure 6 consists of 18 gates. The area overhead is about 4.4% of the original circuit.

Alternating inputs x, $\bar{x}$ are submitted to the circuitry of Figure 6. The outputs $h_1(x), h_2(x), h_3(x)$ are alternating as long as no error occurs.

Now we describe how an arbitrarily given combinational circuit can be actually transformed into a self-dual duplicated circuit. Let $y_j=f_j(x)$, $x=(x_1,\ldots,x_m)$ for $j=1,\ldots,n$ be the Boolean function implemented by the output y_j of the considered combinational circuit f_c. Then we determine for the inputs x_i, $i=1,\ldots,m$, the m self-dual complements $\delta_{j,i}(x)$ of $f_j(x)$ with the minimal number of ones,

$$\delta_{j,i}(x) = x_i(f_j(x)\oplus\bar{f}_j(\bar{x})) \tag{5}$$

and the m self-dual complements $\widetilde{\delta}_{j,i}(x)$ of $f_j(x)$ with the maximal number of ones,

$$\widetilde{\delta}_{j,i}(x) = x_i \vee (f_j(x)\oplus f_j(\bar{x})). \tag{6}$$

All these $2m$ self-dual complements $\delta_{j,i}(x)\widetilde{\delta}_{j,i}(x)$, $i=1,\ldots,m$, are optimized by the SIS system *script.rugged* [17] and the necessary area, for their implementation, is determined.

We choose, from this set of $2m$ self-dual complements of the output y_j, the minimal self-dual complement $\delta_{j,\min}$ of this output with

$$A(\delta_{i,j}) \geq A(\delta_{j,\min}), \quad A(\widetilde{\delta}_{i,j}) \geq A(\delta_{j,\min}).$$

In a first approach the minimal complements $\delta_{1,\min},\ldots,\delta_{n,\min}$ are jointly implemented as a combinational circuit δ. Then the circuit δ and the original circuit f_c are jointly implemented and optimized.

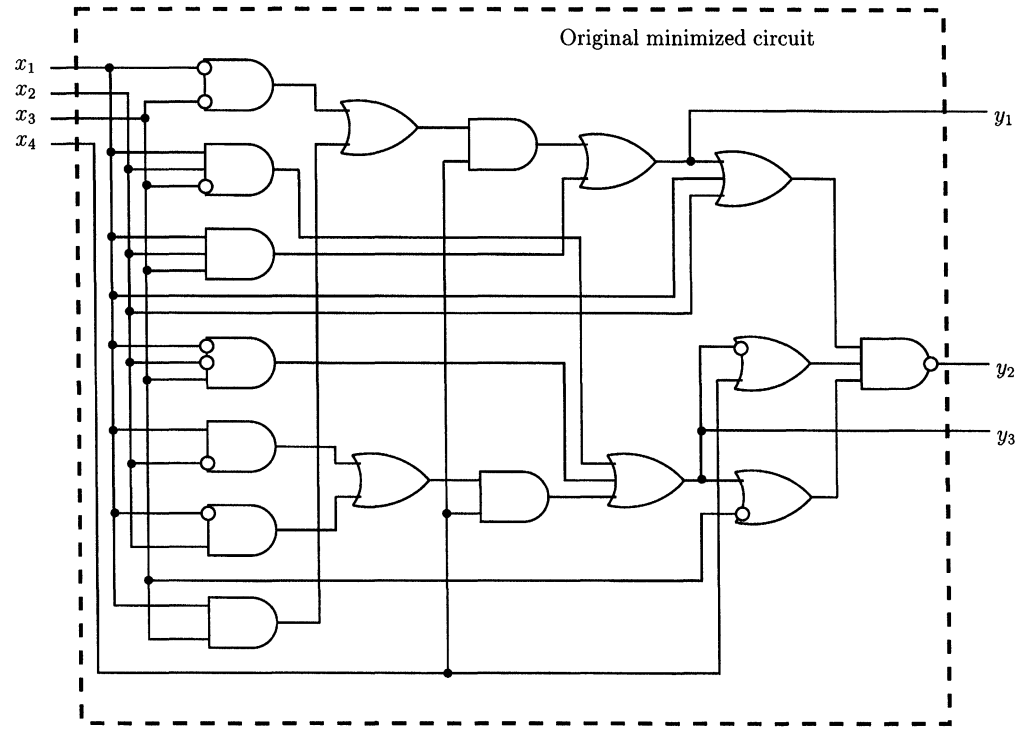

FIGURE 5 Original circuit.

In a second approach the minimal self-dual complements $\delta_{1,\min}, \ldots, \delta_{n,\min}$ of the outputs $y_1, \ldots, y_n$ and the original circuit f_c are jointly optimized in one step. Finally, the minimal circuit obtained in both approaches is chosen.

To model the error detection capability of the circuit in on-line detection mode pseudorandom inputs are used. A sequence of 1000 pseudorandom m-bit inputs is generated. From this sequence a sequence of 2000 inputs consisting of 1000 alternating pairs of inputs is determined. Now the probability not to detect an error at the outputs of the original circuit due to a single stuck-at fault is determined. For every single stuck-at fault ϕ the sequence of 1000 alternating pairs of inputs is submitted to the faulty circuit $f(\phi)$. Let $N(\phi)$ be the number of input pairs for which at least one of the outputs $f_j(\phi)$ of $f(\phi)$ is erroneous under input x or $\bar{x}$ respectively.

Let $n(\phi)$ be the number of input pairs of the sequence of pseudorandom input pairs for which the fault ϕ will be detected by the considered method. Then

$$\mu(\phi) = \frac{N(\phi) - n(\phi)}{N(\phi)} \cdot 100\%$$

is the probability (in percentage), in on-line mode, not to detect an error due to the fault ϕ. For a given circuit for all single stuck-at faults $\phi, j = 1, \ldots, k$, the average value $\bar{\mu}$,

$$\bar{\mu} = \frac{1}{k} \sum_{j=1}^{k} \bar{\mu}(\phi_j)$$

is experimentally determined.

The experimental results for fault coverage for a deterministic test are described now. A test set T for all single stuck-at-0/1 faults is determined by use of the *atpg-script* of SIS [17]. For every input $x \in T$ also the alternating input $\bar{x}$ is submitted to the circuit under test. A fault is detected if, at least

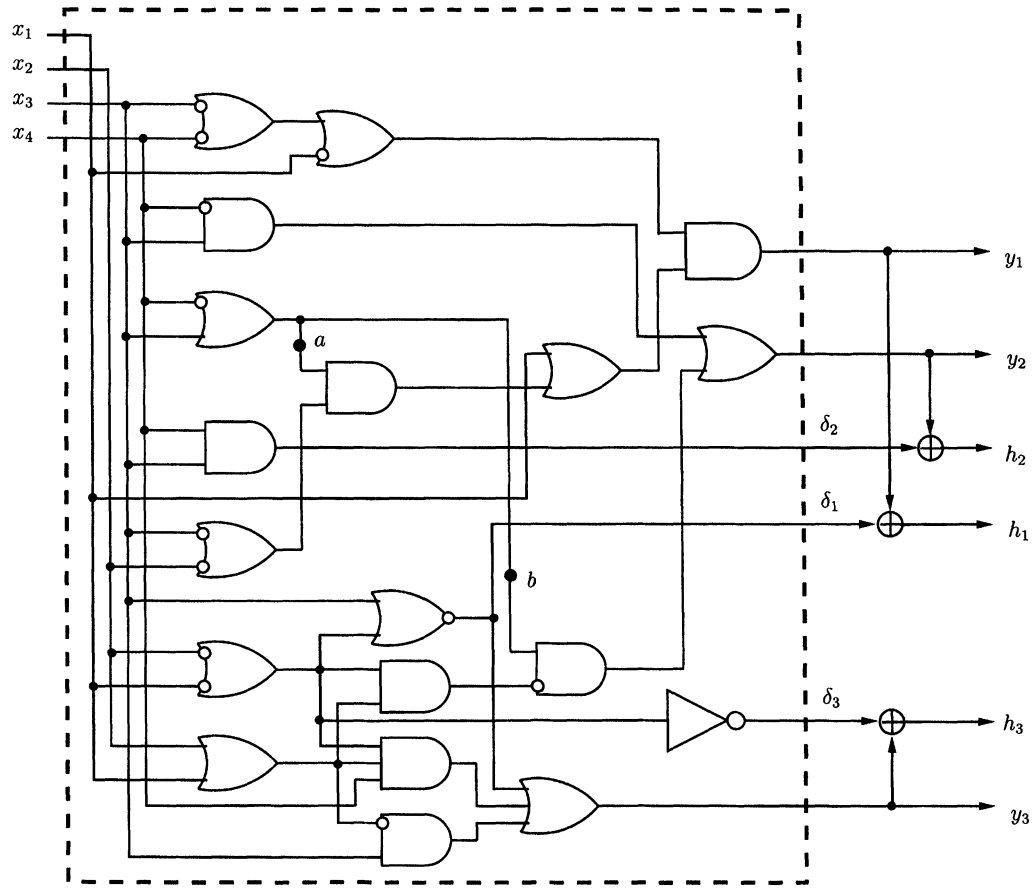

FIGURE 6 Joint implementation of the original circuit and the self-dual complements.

for one of the n circuit outputs y_j, $j=1,\ldots,n$, the value of h_j is not alternating.

For 19 benchmark circuits experimental results are given in Table II. The columns 1, 2 and 3 describe the name of the circuit, the number of inputs and outputs and the area of the optimized original circuit.

For a separate implementation of the original circuit and the corresponding self-dual complements the columns 4 through 8 contain the area overhead for the implementation of the self-dual complements in % of the original optimized circuit, the fault coverage for internal stuck-at faults in %, the probability $\bar{\mu}$ not to detect an erroneous output due to an internal single stuck-at fault, the fault coverage for stuck-at faults of the input lines and the probability $\bar{\mu}$ not to detect an erroneous output caused by a stuck-at fault at the input lines.

The corresponding values for a joint implementation of the original circuit and its self-dual complements are given in column 9 to 13.

The average area overhead, in percentage of the area of the optimized functional circuit, is about 64% for a separate implementation and about 49% for a joint implementation. In the case of a joint implementation, about half of the circuits have an area overhead less than 20% of the optimized area of the original circuits. The average value for the area overhead for these 50% of the circuits is only 7%. Since the circuits cm82a, rd73, b1 and z4ml are self-dual no self-dual complement has to be added to the original circuits.

TABLE II Self-dual duplication. Area overhead, fault coverage and $\bar{\mu}$ for the internal and input faults

Optimal circuit			Separate implementation					Joint implementation				
				int.fault		inp. fault			int. fault		inp. fault	
name	i/o	opt. area	area over. %	fault cover. %	$\bar{\mu}$ %	fault cover. %	$\bar{\mu}$ %	area over. %	fault cover. %	$\bar{\mu}$ %	fault. cover. %	$\bar{\mu}$ %
1	2	3	4	5	6	7	8	9	10	11	12	13
5xp1	7/10	170	25.3	97.9	3.0	100.0	0.0	9.4	98.9	2.8	100.0	0.0
bw	5/28	268	90.3	100.0	1.7	100.0	0.0	73.5	99.6	5.0	100.0	0.0
cm138a	6/8	36	144.0	100.0	0.0	100.0	14.0	131.0	96.6	22.8	100.0	13.9
cm42a	4/10	38	108.0	100.0	0.0	100.0	0.0	92.1	98.8	7.2	100.0	0.0
cm82a	5/3	46	0.0	100.0	0.0	100.0	0.0	0.0	100.0	0.0	100.0	0.0
cm85a	11/3	78	57.7	100.0	0.0	100.0	28.1	12.8	93.9	33.6	100.0	29.1
cmb	16/4	79	29.1	100.0	7.1	81.3	10.0	16.2	97.3	4.7	89.7	2.0
decod	5/16	77	94.7	100.0	0.0	100.0	0.0	93.3	95.8	31.8	100.0	0.0
rd73	7/3	149	0.0	100.0	1.9	100.0	0.0	0.0	100.0	1.5	100.0	0.0
x2	10/7	71	101.0	100.0	0.0	100.0	19.3	64.8	97.1	15.2	100.0	20.5
z4ml	7/4	73	0.0	100.0	0.0	100.0	0.0	0.0	100.0	0.0	100.0	0.0
cu	14/11	90	92.2	100.0	2.1	100.0	12.2	60.0	93.5	28.2	100.0	12.2
ldd	9/19	123	91.7	97.6	4.9	100.0	15.7	83.3	97.9	9.5	100.0	0.0
misexl	8/7	95	73.7	100.0	1.3	100.0	0.0	53.7	98.5	8.6	100.0	0.0
clip	9/5	176	87.5	100.0	1.4	100.0	15.8	85.2	100.0	6.4	100.0	16.2
bl	3/4	17	0.0	100.0	0.0	100.0	17.8	0.0	100.0	0.0	100.0	16.8
con1	7/2	36	108.0	100.0	0.0	85.7	33.1	77.7	94.6	24.7	85.7	33.6
f51m	8/8	156	10.3	98.6	2.4	100.0	0.0	10.9	98.7	1.2	100.0	0.0
pm1	16/13	70	106.0	97.5	4.5	81.3	37.7	62.9	90.4	27.2	86.2	33.9
on the average			64.2	99.5	1.59	97.3	10.2	48.8	97.5	12.1	97.9	9.38

For internal stuck-at faults the average fault coverage in test mode is 99.5% for a separate implementation and 97.5% for a joint implementation. For stuck-at faults at the input lines the average fault coverage is 97.3% for a separate implementation and 97.5% for a joint implementation.

In On-line mode the average probability $\bar{\mu}$ not to detect an error due to an arbitrary internal stuck-at fault is 1.59% for a separate implementation and 12.1% for a joint implementation.

For stuck-at faults at the input lines the corresponding average values for $\bar{\mu}$ are 10.2% for a separate implementation and 9.38% for a joint implementation.

SELF-DUAL PARITY PREDICTION

Self-dual parity checking is a modification of ordinary parity checking. The parity prediction function f_p of ordinary parity checking is replaced by the self-dual complement δ_p of this function such that the modulo-2 sum of the outputs of the monitored circuit and of δ_p is an arbitrary self-dual Boolean function h.

We suppose that the monitored combinational circuit f_c implements the n Boolean functions $y_1=f_1(x)$, $y_2=f_2(x),\ldots,y_n=f_n(x)$. The parity prediction function $f_p(x)$ is defined by

$$f_p(x)=f_1(x)\oplus\cdots\oplus f_n(x). \tag{7}$$

To check the correct behavior of the combinational circuit f by usual parity prediction, the functional outputs $y_1,\ldots,y_n$ are added modulo 2 and compared with the predicted parity $f_p(x)$. If the predicted parity $f_p(x)$ and the modulo 2 sum of the outputs disagree, an error is indicated. (For details and modifications see for instance [15].)

The implementation costs for the parity prediction function are relatively high. Recently, the area

overhead for a joint and a separate implementation was determined in [18, 19] for some benchmark circuits. Although only one additional parity bit $f_p(x)$ is added to the functional circuit f, the additional average area overhead for ordinary parity checking is 44% (29%) of the not optimized original circuit for a separate (joint) implementation of f and $f_p(x)$. The optimization was done by use of the tool SIS [17]. The mentioned results, however, are only achievable for such benchmarks for which an interim representation as a two level expression could be derived by the SIS tool [17]. For other circuits, the area overhead for the implementation of a parity bit may be significantly higher. This large area overhead partially results from the fact that, for a given completely defined circuit f, the parity prediction function $f_p(x)$ is completely determined by Eq. (7); therefore, no don't-care conditions can be utilized to optimize $f_p(x)$.

To reduce the area overhead of the parity bit, we replace the parity prediction function $f_p(x)$ by the self-dual complement δ_p of the parity prediction function $f_p(x)$. δ_p is easier to implement than $f_p(x)$. Instead of checking the circuit by comparing the parity of the outputs with the predicted parity, we substitute the parity prediction function with a self-dual complement of this function. The parity of the outputs and the self-dual complement of the parity prediction function are added modulo 2 to form an arbitrary self-dual Boolean function. The self-duality of this function is constantly monitored by applying alternating inputs. Since we suppose the input variables to be alternating, the output $y = h(x)$ is also alternating as long as no error occurs.

It is easy to see that $\delta'_p(x) = f_p(x) \oplus x_i$ for $i = 1, \ldots, m$ is always a self-dual complement of $f_p(x)$, since $f_p(x) \oplus \delta'_p(x) = x_i$ is self-dual. Thus, the overhead for the optimal self-dual complement $\delta_p(x)$ is, at most, as large as the necessary overhead for the parity prediction function $f_p(x)$.

Figure 7 shows an example circuit with 5 circuit outputs which is monitored by self-dual parity prediction.

The following Boolean functions $f_1, \ldots, f_5$ are implemented in Figure 7.

$$y_1 = f_1(x) = \bar{x}_1 x_2 x_3 \bar{x}_4 \vee (x_1 \vee x_2) x_3 x_4 \vee \bar{x}_1 \bar{x}_3 x_4$$
$$y_2 = f_2(x) = \overline{x_1 \vee x_2} \vee x_1 x_3 \bar{x}_4 \vee x_1 \bar{x}_3 x_4$$
$$y_3 = f_3(x) = \bar{x}_2 x_2 x_3 \bar{x}_4 \vee \overline{(x_3 \vee x_4)} \vee \bar{x}_1 \bar{x}_2 x_3$$
$$y_4 = f_4(x) = \bar{x}_1 \bar{x}_2 x_3 \vee \bar{x}_1 \bar{x}_3 x_4 \vee x_1 \bar{x}_3 x_4 \vee x_1 x_3 \bar{x}_4$$
$$y_5 = f_5(x) = \overline{x_3 \vee x_4} \vee (x_3 \vee x_4) \bar{x}_1 x_2.$$

The parity bit is determined as

$$y_p = f_p(x) = f_1(x) \oplus \cdots \oplus f_5(x)$$
$$= \bar{x}_1 \bar{x}_2 \vee \bar{x}_1 \bar{x}_3 x_4 \vee \bar{x}_1 x_3 \bar{x}_4 \vee x_1 x_3 x_4$$

and a possible self-dual complement $\delta_p(x)$ of the parity prediction function $f_p(x)$ is

$$\delta_p(x) = x_2 (f_p(x) \oplus \bar{f}_p(\bar{x})) = x_2 x_3 x_4.$$

The complexity of the self-dual complement $\delta_p(x)$, consisting only of 2 two-input gates, is very small. A fault will be detected if, in the presence of the fault, the parity is erroneous for some input x but correct under input $\bar{x}$. The experimental results show that the overhead for the self-dual complement $\delta_p(x)$ is much smaller than the necessary overhead for the parity prediction function $f_p(x)$.

Experimental results are presented in Table III. As in Table II columns 1, 2 and 3 of Table III describe the name of the circuit, the number of inputs and outputs and the area of the optimized original circuit. The area overhead in percentage of the original optimized circuit, the fault coverage (in percentage) for internal single stuck-at faults in test mode, the probability $\bar{\mu}$ (in percentage) not to detect in normal operation mode an erroneous circuit output due to an internal single stuck-at fault, the fault coverage (in percentage) for single stuck-at faults at the input lines and the probability $\bar{\mu}$ (in percentage) not to detect in normal operation mode an erroneous circuit output due to a single stuck-at fault at the input lines are given in columns 4 through 8 for a separate implementation of the original circuit and the self-dual complement of the parity function.

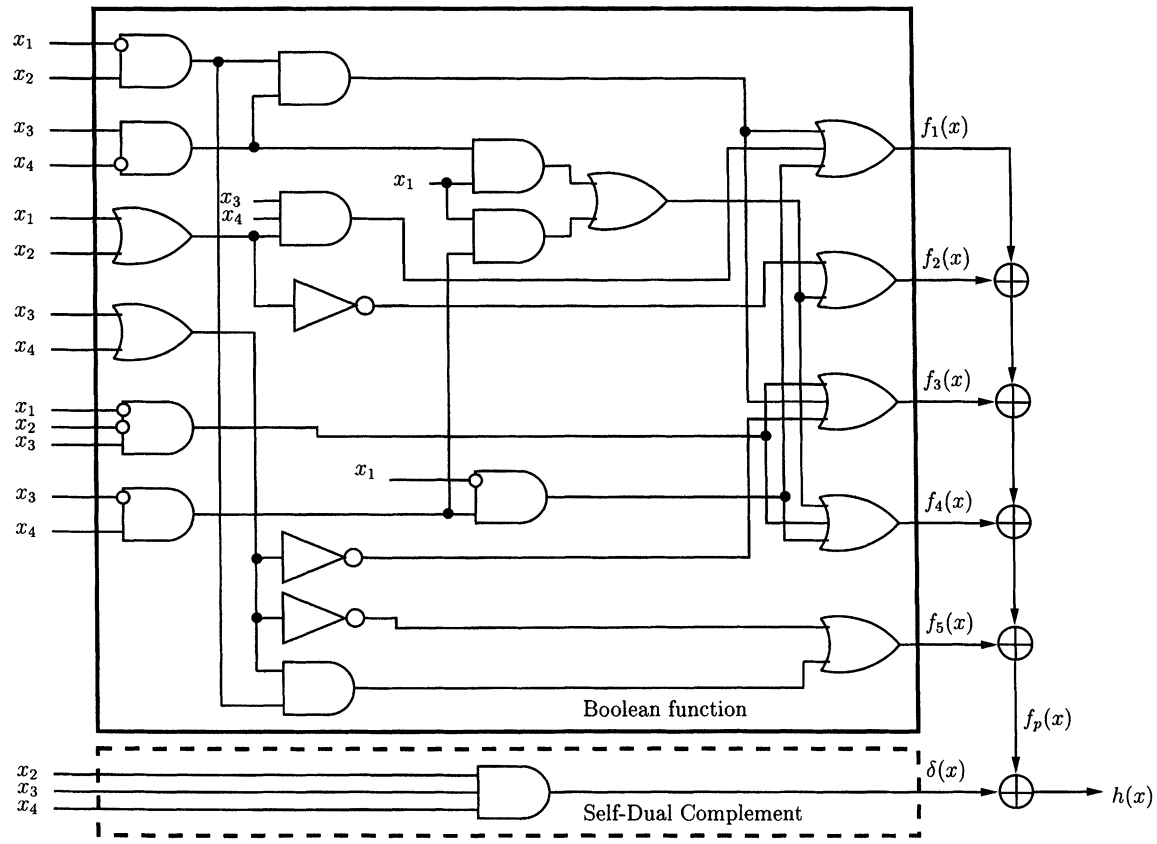

FIGURE 7 Monitoring by self-dual parity prediction.

For a joint implementation the corresponding results are presented in columns 9 to 12 of Table III. The average value for the area overhead for the implementation of self-dual parity is 27% of the optimized circuit for a separate implementation and 18% for a joint implementation. For the considered benchmark circuit the corresponding values for ordinary parity prediction are 62% for a separate implementation and 28.9% for a joint implementation. (In the literature the area overhead for parity prediction is sometimes given in percentage of the area of the not optimized circuits.) For self-dual parity in test mode the average values for fault coverage with respect to internal single stuck-at faults are 93.3% for a separate implementation and 87.2% for a joint implementation. For ordinary parity prediction the corresponding values are 96.1% for a separate implementation and 83.5% for a joint implementation. For a separate implementation fault coverage for self-dual parity is only 2% less than for ordinary parity prediction, and the area overhead is only 43% of the area overhead for ordinary parity prediction. For a joint implementation the fault coverage is even higher for self-dual parity.

For self-dual parity the average probability $\bar{\mu}$ not to detect an error due to an internal single stuck-at fault is 22.7% for a separate implementation and 33.1% for a joint implementation. For ordinary parity prediction the corresponding values are 22.4% and 42.6%. These probabilities are relatively high for both ordinary parity prediction and self-dual parity. For single stuck-at faults at the input lines the average fault coverage is 73.4% for a separate implementation and 73.9% for a joint implementation. For the

TABLE III Experimental results for parity and self-dual parity

Optimal circuit			Separate implementation					Joint implementation				
				int. fault		inp. fault			int. fault		inp. fault	
name	i/o	opt. area	area over. %	fault cover. %	$\bar{\mu}$ %	fault cover. %	$\bar{\mu}$ %	area over. %	fault cover. %	$\bar{\mu}$ %	fault cover. %	$\bar{\mu}$ %
1	2	3	4	5	6	7	8	9	10	11	12	13
5xp1	7/10	170	28.8	96.1	19.6	100.0	53.6	18.8	93.7	32.5	100.0	53.9
bw	5/28	268	6.8	100.0	27.4	100.0	52.8	3.4	92.0	39.7	100.0	53.8
cm138a	6/8	36	13.9	100.0	0.0	50.0	66.7	13.9	100.0	0.0	50.0	66.7
cm42a	4/10	38	10.1	100.0	0.0	75.0	53.8	7.9	94.9	16.2	75.0	53.6
cm82a	5/3	46	0.0	89.8	21.7	100.0	35.6	0.0	75.0	34.3	100.0	36.9
cm85a	11/3	78	59.7	100.0	31.9	27.3	72.7	11.6	58.4	63.2	27.3	72.7
cmb	16/4	79	32.9	85.6	42.9	6.3	93.3	17.7	54.3	100.0	6.9	100.0
decod	5/16	77	0.0	98.8	0.9	20.0	80.0	0.0	99.5	0.9	20.0	80.0
rd73	7/3	149	0.0	82.4	24.1	100.0	20.4	0.0	83.1	34.6	100.0	19.1
x2	10/7	71	36.6	95.8	24.1	88.9	59.3	30.9	90.9	35.3	84.2	59.9
z4ml	7/4	73	0.0	83.5	32.8	100.0	42.1	0.0	82.4	31.2	100.0	41.0
cu	14/11	90	44.0	90.0	29.4	35.7	82.0	22.9	78.4	52.2	41.7	78.0
ldd	9/19	123	12.2	92.6	34.2	80.0	53.0	8.1	91.1	41.0	80.0	53.3
misexl	8/7	95	27.5	91.2	33.9	50.0	67.1	25.0	91.7	30.5	50.0	66.5
clip	9/5	176	53.4	98.0	20.4	100.0	34.1	48.9	96.9	28.1	100.0	34.7
bl	3/4	17	0.0	83.3	29.7	100.0	49.7	0.0	82.6	2.4	100.0	52.0
con1	7/2	36	88.9	100.0	5.2	100.0	46.8	80.6	98.7	23.9	100.0	47.5
f51m	8/8	156	9.0	98.6	29.3	100.0	46.7	6.5	96.0	29.2	100.0	46.7
pm1	16/13	70	92.0	98.6	23.8	62.1	59.4	47.1	97.3	34.0	69.2	54.6
on the average			27.1	93.9	22.7	73.4	56.3	18.1	87.2	33.1	73.9	56.4

circuits cm82a, decod, rd73, z4ml and b1 the parity function is self-dual and no self-dual complement has to be added.

The probability $\bar{\mu}$ in on-line mode not to detect a functional error due to a single stuck-at fault at the input lines is 56% for both realizations.

If an error due to a permanent fault occurs at the circuit outputs n times, it will not be detected with a probability of only $\bar{\mu}^n$ which already is a very small quantity for relatively small values of n. But the probability to miss an error resulting from a transient fault remains relatively high.

LINEAR OUTPUT SPACE COMPACTION OF ALTERNATING SIGNALS

In this section linear space compaction of alternating signals will be considered. The alternating outputs $g_1, \ldots, g_n$ of the self-dual circuit f_{sd} are compacted by a linear compaction circuit LCC into q alternating signals $v_1, \ldots, v_q$, $1 \leq q < n$. Only the compacted signals $v_1, \ldots, v_q$ are to be monitored, whether they are alternating or not. In [12] the alternating output signals are transformed into two-rail signals by use of D-flip-flops and then monitored by a two rail checker with $2n$ inputs and 2 outputs. The linear output space compaction for alternating signals as described in this section reduces the necessary area overhead for output checking by a factor of 2.5 to 3.

The linear compaction circuit LCC has to be self-dual and self-testing with respect to every single stuck-at-0/1 fault. Self-testing here means that every single stuck-at-0/1 fault is detectable by a non-alternating signal of at least one of the outputs of the LCC if the appropriate alternating signals are applied to its inputs.

The linear compaction circuit LCC can be realized by use of a linear self-dual module $M(g_1, g_2, g_3)$ with the three inputs g_1, g_2, g_3 and one output implementing the linear function

$M(g_1, g_2, g_3) = g_1 \oplus g_2 \oplus g_3$, in $GF(2)$. This function is self-dual. A direct implementation of the complete disjunctive normal form of M, $M(g_1, g_2, g_3) = \bar{g}_1\bar{g}_2g_3 \vee \bar{g}_1g_2\bar{g}_3 \vee g_1\bar{g}_2\bar{g}_3 \vee g_1g_2g_3$, is self-testing, whereas an implementation of M by two XOR-gates is not.

As an example the linear compaction circuit LCC for the compaction of ten alternating signals $g_1, \ldots, g_{10}$ into one alternating signal v according to a parity code is shown in Figure 8.

The linear compaction circuit LCC is implemented as a linear chain of 5 linear self-dual modules $M_1, \ldots, M_5$. Every module M_i, $i = 1, \ldots, 5$, compacts three alternating signals into one alternating signal. Since 10 is an even number, the binary periodic signal a (the clock signal) is utilized as an additional alternating input for the module M_5 in Figure 8. The compacted signal v represents the self-dual parity signal $v = g_1 \oplus g_2 \oplus \cdots \oplus g_{10} \oplus a$ of the alternating outputs $g_1, \ldots, g_{10}$ of the original self-dual circuit f and the additional periodic signal a. The compacted signal v is determined by a single chain of M-modules.

To improve the fault coverage it is sometimes useful linearly to compact the alternating output signals of the CUT into several different alternating signals. A very high fault coverage is achieved if the outputs are linearly compacted according to a Hamming-code.

As an example the compaction of ten alternating signals $g_1, \ldots, g_{10}$ into four alternating signals v_1, v_2, v_3, v_4 according to a Hamming code is illustrated in Figure 9. Every alternating output v_i, $i = 1, \ldots, 4$, is determined by a linear chain of 3 M-modules. For each of the four alternating output signals the binary periodic signal a is used as an additional alternating input signal to guaranty the necessary three alternating inputs for every module M. The monitored outputs v_i, $i = 1, \ldots, 4$ with

$$v_1 = g_1 \oplus g_2 \oplus g_4 \oplus g_5 \oplus g_7 \oplus g_9 \oplus a,$$
$$v_2 = g_1 \oplus g_3 \oplus g_4 \oplus g_6 \oplus g_7 \oplus g_{10} \oplus a,$$
$$v_3 = g_2 \oplus g_3 \oplus g_4 \oplus g_8 \oplus g_9 \oplus g_{10} \oplus a,$$
$$v_1 = g_5 \oplus g_6 \oplus g_7 \oplus g_8 \oplus g_9 \oplus g_{10} \oplus a$$

are chosen to be the self-dual group parities of a corresponding Hamming code with ten information bits $g_1, \ldots, g_{10}$ and four control bits v_1, v_2, v_3, v_4 and with the parity matrix

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

In general, the self-dual and self-testable LCC with q outputs consists of q linear chains of M-modules. In test mode the average value of not detected internal stuck-at faults is 8.4%, and in normal operation mode the average probability not to detect an error due to an internal stuck-at fault is 22.1% if the self-dual circuit outputs are compacted by a parity code. Output compaction by a Hamming code results only in a negligible deterioration of the fault-coverage in test mode and the error detection probability in normal operation mode.

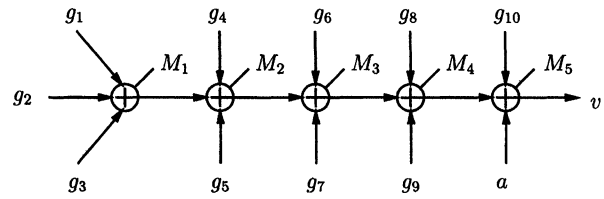

FIGURE 8 Linear compaction circuit LCC.

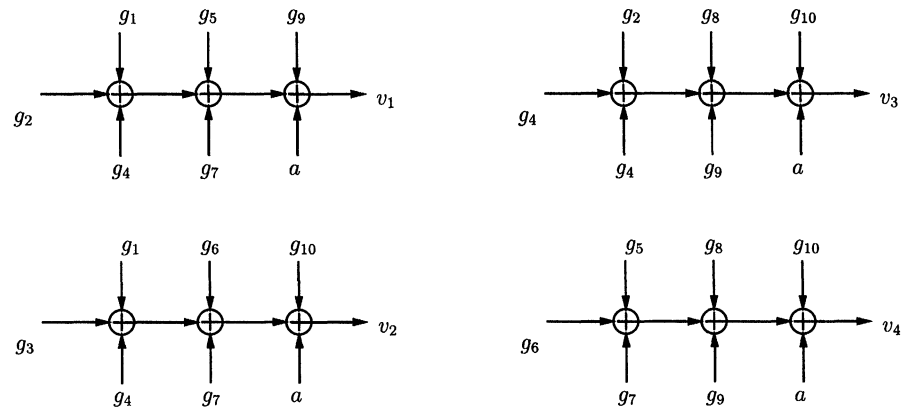

FIGURE 9 The compaction of ten alternating signals into four alternating signals according to a Hamming code.

The additional area overhead for the described linear output data compaction is determined by the number of outputs. Thus the percentage of the additional area overhead is determined by the area A (the number of literals) of the self-dual circuit and by the number |out| of its outputs.

The average area $\overline{\text{area}}$, $\overline{\text{area}} = (A/|\text{out}|)$, for the implementation of a single circuit output of the considered self-dual circuit is a good estimate for the additional area overhead (in percentage of the area of the original circuit). The average values of the additional area overhead $\Delta A_P(\%)$ for a parity code and $\Delta A_H(\%)$ for a Hamming-code in percentage of the area of the original circuit are presented in Table IV. For $\overline{\text{area}} > 40$ for a parity code ΔA_P is less than 20% of the area of the original circuit. The corresponding value ΔA_H for a Hamming code is less than 50% of the area of the original circuit. For $\overline{\text{area}} < 20$ the corresponding values are 50% for a parity code and 100% for a Hamming code respectively.

TABLE IV Area overhead for data compaction

$\overline{\text{area}} = (A/\|\text{out}\|)$	$\Delta A_P(\%)$	$\Delta A_H(\%)$
20	50	100
30	30	75
40	20	50
50	18	30
100	5	15

REDESIGN OF SELF-DUAL CIRCUITS INTO SELF-DUAL FAULT-SECURE CIRCUITS

In this chapter we describe how a self-dual circuit f_{sd} can be transformed into a self-dual fault-secure circuit f_{sdf}. The transformed circuit will be self-dual fault-secure with respect to all single unidirectional gate faults.

If the self-dual circuit is implemented by use of self-dual complements of its functional outputs the original circuit is only fault secure if the original circuit outputs and the self-dual complements of these outputs are separately implemented.

The original self-dual circuit f_c is supposed to be given as a net list of gates, where $G = \{g_1, \ldots, g_N\}$ denotes the set of gates. As we have already pointed out the gates are supposed to be AND-, OR-, NAND- and NOR-gates and inverters. First we determine the gates of the original circuit which are not functionally fault secure with respect to some of the circuit outputs according to Theorem 2. Then we duplicate all the gates which are on paths from these not fault-secure gates to the corresponding outputs. These gates are connected in such a way that all the paths from a non fault-secure gate to the corresponding circuit outputs have the same parity of inverters. Gates which

are not connected to any circuit output are deleted. According to Theorem 2 the circuit is self-dual fault-secure.

The algorithm for the transformation of self-dual circuit f_{sd} into a self-dual fault-secure circuit f_{sdf} is described now in more detail.

ALGORITHM 1

1. We determine the set $G_s = \{g_1^s, \ldots, g_M^s\}$ of self-dual fault-secure gates of f_c.
2. We compute the set $G_n = G \setminus G_s$ of not self-dual fault-secure gates of f_c.
3. If $G_n = \emptyset$ then STOP.
4. For every gate $g_i \in G_n$ we determine the set of outputs $Y(g_i)$ for which g is not self-dual fault-secure.
5. For every gate $g_i \in G_n$, we determine the transitive fanout $T(g_i) \setminus Y(g_i)$ of g_i with respect to $Y(g_i)$. This transitive fanout is the set of gates which is connected by a path from g_i to one of the outputs $Y(g_i)$ of f_c. g_i is considered to be an element of $T(g_i) \setminus Y(g_i)$.
6. We compute the transitive fanout $T(G_n)$ of all not self-dual fault-secure gates with respect to their not self-dual fault-secure outputs

$$T(G_n) = \bigcup_{g_i \in G_n} T(g_i) \setminus Y(g_i).$$

7. All the gates g, $g \in T(G_n)$, are initially duplicated into g^0 and g^1 with an even (0) and odd (1) superscript.
8. The duplicated gates will be connected according to the following rules:
   - 8.1. If for $g \in T(G_n)$ the output of the gate g in f_c is directly connected (*via* an inverter) to a circuit output y_i, then $g^0(g^1)$ in f_{sdf} is directly connected (*via* an inverter) to the output y_i.
   - 8.2. If for $g, h \in T(G_n)$ the output of the gate g is connected to an input of gate h *via* an even (odd) number of inverters, then in f_{sdf} the gate $g^k(g^{k \oplus 1})$ is connected *via* the same number of inverters to the corresponding input of h^k, for $k = 0, 1$.
   - 8.3. If for $g \in G \setminus T(G_n)$ and $h \in T(G_n)$ in f_c the output of g is connected (*via* an even or odd number of inverters) to an input of h, then in f_{sdf} the output of g is connected to the corresponding inputs of both the gates h^0 and h^1.
   - 8.4. If for $g \in G \setminus T(G_n)$ and $h \in T(G_n)$ in f_c the output of h is connected (*via* an even or odd number of inverters) to an input of g, then in f_{sdf} either the output of h^0 or the output of h^1 is connected *via* the same number of inverters to the corresponding input of g.
   - 8.5. If a primary input l in f_c is connected to an input of a gate g, $g \in T(G_n)$ then in f_{sdf} the primary input l is connected to the corresponding inputs of both the gates g^0 and g^1.
9. Gates that are not connected to at least one of the circuit outputs are removed.
10. Redundant gates are identified by an ATPG-tool such as HANNIBAL [20] and removed from the circuit.
11. GOTO 1.

There was only one example for which the loop of the Algorithm was passed twice. This was the case where a gate of the original circuit was functionally self-dual fault-secure with respect to an output y_k since one of its stuck-at faults could not be sensitized to this output (but to another output). After the circuit transformation, all the stuck-at faults of this gate could be sensitized to the output y_k and the gate became not self-dual fault-secure with respect to y_k. In all other cases the Algorithm stopped after the first pass.

The transformation of a given self-dual circuit into a fault-secure self-dual circuit is demonstrated now for the example of Figure 3 described in the second chapter.

The gates $2, 3, 5-10$ are structurally fault secure. Since there is a path (1, 2, 7, 8) with an even (0) number of inverters as well as a path (1, 3, 5, 7, 8) with an odd (1) number of inverters from gate 1 to the circuit output y_1 this gate is not structurally self-dual fault-secure with respect to the output

y_1. Also, gate 4 is not structurally self-dual fault-secure with respect to the output y_1. Both these gates are structurally self-dual fault-secure with respect to the output y_2. According to Theorem 2, both these gates are also not functionally self-dual fault-secure with respect to the output y_1.

Now we transform the self-dual circuit of Figure 3 into a self-dual fault-secure circuit using the proposed algorithm 1. According to the steps of the described algorithm we obtain:

1. $G_s = \{2, 3, 5-10\}$;
2. $G_n = \{1, 4\}$;
3. $G_n \neq \emptyset$;
4. $Y(1) = \{y_1\}$, $Y(4) = \{y_1\}$;
5. $T(1) \setminus Y(1) = \{1-3, 5, 7, 8\}$, $T(4) \setminus Y(4) = \{4, 6, 8\}$;
6. $T(G_n) = \{T(1) \setminus Y(1)\} \cup \{T(4) \setminus Y(4)\} = \{1-8\}$;
7. All the gates of $T(G_n)$ are duplicated into gates $\{1^0, 1^1, 2^0, 2^1, \ldots, 8^0, 8^1\}$ with superscript 0 and superscript 1. The original gates $\{1, 2, \ldots, 8\}$ are removed;
8. The duplicated gates are connected as shown in Figure 10;
9. The gates $\{2^1, 3^0, 5^0, 6^1, 7^1, 8^1\}$ are not connected to a circuit output and are therefore deleted. As a result only the gates 1 and 4 are finally duplicated;
10. The circuit contains no redundant gates.

Since all the gates of the circuit in Figure 10 are structurally self-dual fault-secure, G_n is now empty and a second pass of the algorithm is not needed.

To obtain experimental results 34 benchmark circuits were transformed into self-dual circuits [21]. All redundancies were removed by the ATPG-tool HANNIBAL [20]. Then these circuits were transformed into self-dual fault-secure circuits according the proposed algorithm and again all redundancies were removed. The average area overhead for the transformation of a self-dual circuit into a self-dual fault-secure circuit is 16.98%. For 82% of the benchmark circuits the area overhead is less than 45%. For these 82% of the

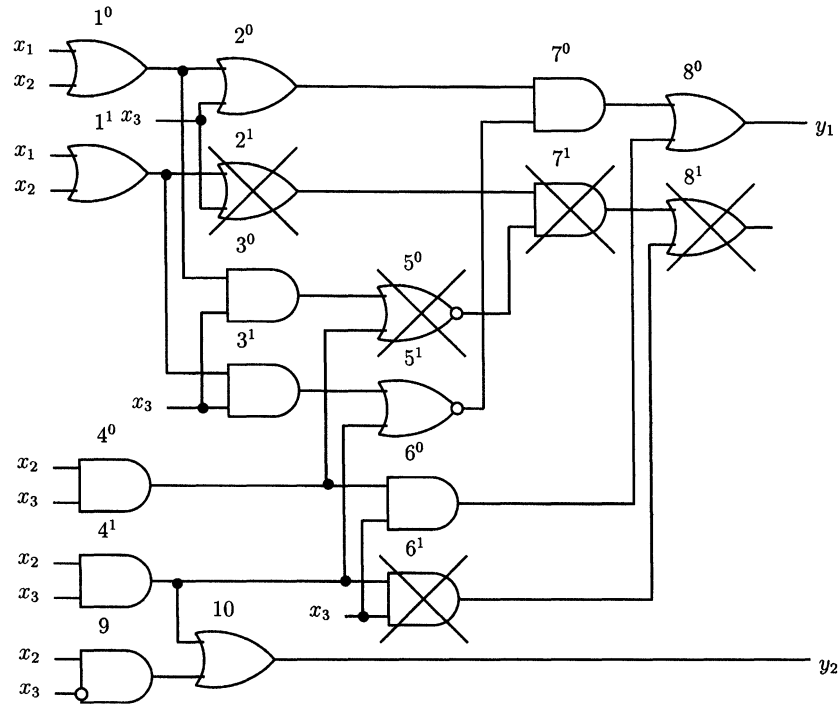

FIGURE 10 Self-dual fault-secure circuit for the transformation of a self-dual circuit into a self-dual fault secure circuit.

benchmark circuits the average area overhead is only 8.81%.

CONCLUSIONS

In this paper new methods for the design of self-dual circuits with alternating inputs and their application for error detection and testing were presented. These methods are based on the new concept of a self-dual complement of a Boolean function. The proposed synthesis methods utilize the different self-dual complements either with a minimal number of ones or with a maximal number of ones. Self-dual duplication and self-dual parity prediction were considered in detail. Contrary to ordinary duplication and comparison the functional circuit and the corresponding self-dual complement can be jointly implemented. The differences between joint and separate implementations for self-dual duplication and self-dual parity prediction were experimentally investigated. In general, a joint implementation instead of a separate one results in a large reduction of the necessary area overhead, a small decrease of the fault coverage in test mode, and a somewhat more significant increase of the probability not to detect an error in on-line mode. Linear self-testing space compaction of alternating signals can be used to reduce the number of alternating outputs which are to be monitored. For safety-critical application the concept of self-dual fault-secureness was introduced. It was shown how a given self-dual circuit can be easily transformed into a self-dual fault-secure circuit.

In on-line mode the original functional inputs x and the corresponding inverted inputs $\bar{x}$ are always subsequently submitted to the self-dual circuit and a 100% time redundancy is necessary in this mode. Therefore the proposed method is mainly useful for control systems for which time is not a critical issue. In On-line mode also stuck-at faults of the input lines are detected with a high probability. In test mode the elements x_t of a test set T and their inverted values $\bar{x}_t$ are applied to the circuit under test. A fault is detected if the corresponding test responses are not alternating. For that reason it is not necessary to store the test responses. In fast mode without error detection only the functional inputs are applied to the circuit.

It was demonstrated that the design of self-dual circuits by use of self-dual complements is useful for error detection and testing. The combination of the different modes of operation allows to achieve different levels of error detection and fault tolerance.

Acknowledgments

The authors are grateful to Prof. E. Sogomonyan, for stimulating discussions concerning the separate and joint implementations of error detection circuits. This paper was supported by the Deutsche Forschungsgemeinschaft, grant Go536/4-2 and by the Federal Government of Germany, project WTZ, RUS 121-96.

References

[1] Sellers, F., Hsiao, M. and Bearnson, L., *Error Detecting Logic for Digital Computers*, New-York: McGraw-Hill, 1968.

[2] Rao, T. R. N. and Fujiwara, E., *Error Control Coding for Computer Systems*, New-York: Prentice Hall, 1989.

[3] Nicolaidis, M., Noraz, S. and Courtois, B., "A generalized theory of fault-safe systems", In: *19th IEEE International Symposium on Fault Tolerant Computing*, pp. 398–406, IEEE Press, 1989.

[4] Johnson, B. W., *Design and Analysis of Fault Tolerant Systems*, Addison-Wesley, 1989.

[5] Jha, N. K. and Wang, S.-J., "Design and synthesis of self-checking vlsi circuits and systems", *IEEE Transactions on Computer Aided Design*, pp. 878–887, June, 1993.

[6] De, K., Natarajan, C., Nair, D. and Banerjee, P., "Rsyn: A system for automated of reliable multilevel circuits", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, pp. 186–195, June, 1994.

[7] Touba, A. and McCluskey, E. J. (1994). "Logic synthesis techniques for reduced area implementation of multilevel circuits with concurrent error detection", In: *International Conference on Computer Aided Design*, pp. 651–654.

[8] Lala, P. K., *Fault Tolerant and Fault Testable Hardware Design*, Englewood Cliffs, N.J. Prentice-Hall, 1985.

[9] Minero, R., Anello, A., Furey, R. and Palounek, L. R. (1972). "Checking by pseudoduplication", *Tech. Rep.*, US Pat. 3660646, G06F, 11/00.

[10] Patel, J. and Fung, L. (1982). "Concurrent error detection in ALUs by recomputing with shifted operands", *IEEE Trans. Comp.*, **C-31**, 589–595.

[11] Patel, J. and Fung, L. (1983). "Concurrent error detection in multiply and divide arrays", *IEEE Trans. Comp.*, **C-32**, 417–422.
[12] Reynolds, D. A. and Metze, G. (1978). "Fault detection capabilities of alternating logic", *IEEE Transactions on Computers*, **C-27**(12), 1093–1098.
[13] Moshanin, V., Saposhnikov, Vl., Saposhnikov, V. and Gössel, M., "Synthesis of self-dual multi-output combinational circuit for on-line testing", In: *2nd IEEE International On-line Testing Workshop* (Saint-Jean de-Luz, Biarritz, France), pp. 107–11, July, 1996.
[14] Saposhnikov, Vl., Dmitriev, A., Saposhnikov, V. and Gössel, M. (1996). "Self-dual parity checking – a new method for on-line testing", In: *14th IEEE International VLSI Test Symposium* (Princeton, N.J.), pp. 162–168.
[15] Gössel, M. and Graf, S., *Error Detection Circuits*, New-York: McGraw-Hill, 1993.
[16] Saposhnikov, V., Saposhnikov, Vl., Dmitriev, A. and Gössel, M., "Self-dual duplication for error detection", In: *Proc. 7th Asian Test Symposium* (Singapore), pp. 296–300, IEEE Computer Society, December, 1998.
[17] Sentovich, E., Sing, K., Lavagno, I., Moon, C., Murgai, R., Saldanha, A., Savoj, H., Stephan, P., Brayton, R. and Sangiovanni-Vincentelli, A. (1992). "Sis: A system for sequential circuits synthesis", *Tech. Rep.*, University of Berkeley.
[18] Gupta, S. and Pradhan, D. (1992). "Can concurrent chekers help BIST?", In: *Proceedings 1992 International Test Conference*, pp. 140–150, IEEE Computer Society.
[19] Sogomonyan, A. E. (1994). "Concurrent error-detection circuit evaluation", *Tech. Rep.*, Max-Planck-Society, University of Potsdam.
[20] Kunz, W. (1993). "Hannibal: An efficient tool for logic verification based recursive learning", In: *International Conference on Computer Aided Design (ICCAD)*, pp. 538–543.
[21] Morosov, A., Saposhnikov, V., Saposhnikov, Vl. and Gössel, M., "Design of self-dual fault-secure combinational circuit", In: *3rd IEEE International On-line Testing Workshop*, pp. 233–237, July, 1997.

Authors' Biographies

Alexej Dmitriev has studied at the Higher Institute of Railway Transportation in St. Petersburg, Russia from 1989–1994 and graduated as an engineer in 1994. Since 1995 he is a graduate student in the Fault-Tolerant Computing Group at the University of Potsdam. He is working in the area of self-checking and self-testing circuits where he has published 5 papers.

Valerij Saposhnikov is a pro-rector for research at the University of Transportation, St. Petersburg, Russia since 1989. He received the Doctor of Science degree in 1980 from this University, where he is also a lecturer and a full professor of discrete systems theory. He teaches and does research in the fields of automata theory. He is an author of numerous scientific papers and co-author of 8 books (together with Vladimir Saposhnikov). He is a member of the Russian Transportation Academy.

Vladimir Saposhnikov is a head of the Department for Automation and Control at the University of Transportation, St. Petersburg, Russia since 1991. He received the Doctor of Science degree in 1984 from this University, where he was also a lecturer and a full professor for Computer Science. He teaches and does research in the fields of automata theory, computer aided design of testable computers and self-checking system. He is an author of numerous scientific papers and the co-author of 8 books (together with Valerij Saposhnikov). He is a member of the Russian Transportation Academy.

Michael Goessel received the Dr. rer. nat. degree in Physics from the University of Jena and the Dr. sc.nat. degree in Computer Science from the Technical University of Dresden, Germany. From 1969 to 1991 he was with the Institute of Cybernetics of the Academy of Sciences in Berlin. From 1992 to 1996 he was the head of the Fault-Tolerant Computing Group of the Max-Planck-Society at the University of Potsdam. Since 1994 he is also a full professor of Informatics at the University of Potsdam, Germany. He has published over 120 research papers in the fields of automata theory, non-linear systems, parallel memories, image processing and fault-tolerant computing. He is the author and co-author of 7 books, including Error Detection Circuits (with S. Graf), McGraw-Hill, 1993 and Memory Architecture and Parallel Access (with B. Rebel and R. Creutzburg), Elsevier, 1994.

Vladimir Moshanin has studied at the Higher Institute of Railway Transportation in St. Petersburg, Russia from 1987–1992 and graduated as an engineer in 1992. After that he has worked in the Laboratory of Railway Safety Systems at this Institute. Since 1994 he is a graduate student in the Fault-Tolerant Computing Group at the University of Potsdam. He is working in the area of

self-checking and self-testing circuits where he has published 5 papers.

Andrej Morosov has studied at the Higher Institute of Railway Transportation in St. Petersburg, Russia from 1987–1992 and graduated as an engineer in 1992. Since 1992 he is a graduate student in the Fault-Tolerant Computing Group at the University of Potsdam. He received the Doctor rer.nat. degree in 1998 from the University of Potsdam. He is working in the area of self-checking and self-testing circuits where he has published 10 papers.