A Minimal Universal Test Set for Self-Test of EXOR-Sum-of-Products Circuits

Ugur Kalay, Douglas V. Hall, Marek A. Perkowski

Abstract—A testable EXOR-Sum-of-Products (ESOP) circuit realization and a simple, universal test set which detects all single stuck-at faults in the internal lines and the primary inputs/outputs of the realization are given. Since ESOP is the most general form of AND-EXOR representations, our realization and test set are more versatile than those described by other researchers for the restricted GRM, FPRM, and PPRM forms of AND-EXOR circuits. Our circuit realization requires only two extra inputs for controllability and one extra output for observability. The cardinality of our test set for an n input circuit is (n+6). For Built-in Self Test (BIST) applications, we show that our test set can be generated internally as easily as a pseudo-random pattern, and that it provides 100% single stuck-at fault coverage. In addition, our test set requires a much shorter test cycle than a comparable pseudo-exhaustive or pseudo-random test set.

Index Terms—Universal test set, AND-EXOR realizations, Reed-Muller expressions, single stuck-at fault model, easily testable combinational networks, Design for Testing (DFT), self testable circuits, Built-in Self Test (BIST), test pattern generation.

1 INTRODUCTION

The large increase in the complexity of ASICs has led to a much greater need for circuit testability and Built-In-Self-Test (BIST) [1]. The testability properties of different forms of two-level AND-EXOR networks have attracted many researchers [2, 3, 4, 5, 6]. The forms investigated include Positive Polarity Reed-Muller (PPRM) [2], Fixed Polarity Reed-Muller (FPRM) [6], Generalized Reed-Muller (GRM) [7], and EXOR-Sum-of-Products (ESOP) [3]. All of the canonical Reed-Muller forms (PPRM, FPRM, and GRM) have restrictions on the allowed polarities of variables or on the allowed product terms. ESOP, on the other hand, has no restrictions, and is formed by combining arbitrary product terms using EXORs [3, 7]. Therefore, ESOP is the most general form of 2-level AND-EXOR networks.

The Reed-Muller expansion of an arbitrary function is: $f = c_0 \oplus c_1 x_1^* \oplus c_2 x_2^* \oplus \cdots \oplus c_n x_n^* \oplus c_{n+1} x_1^* x_2^* \oplus \cdots \oplus c_{2n-1} x_1^* x_2^* \cdots x_n^*$, where x_n^* is a literal term that can be a variable x_n or its negation $\overline{x}_n$, and c_n is a constant term that can be '0' or '1'. PPRM, also called the Reed-Muller form, is the most restricted form in that only positive polarities are allowed for input variables. FPRM allows only one polarity for each input variable. GRM has no restrictions on the allowed polarities of variables but does not allow the same set of variables in more than one product term. For example, $f = 1 \oplus x_1 \oplus x_1 x_2 \oplus x_2 x_3$ is a PPRM since the variables appear with only positive polarities in the expression. $f = x_1 \oplus x_1 \overline{x}_2 \oplus \overline{x}_2 x_3$ is an FPRM because negative polarity exists and each variable appears with only one polarity; either negative or positive. $f = \overline{x}_1 x_2 \oplus \overline{x}_2 x_3 \oplus \overline{x}_1 x_2 x_3$ is a GRM because the variable x_2 appears with both positive and negative polarities. $f = x_1 x_2 x_3 \oplus \overline{x}_1 \overline{x}_2 \overline{x}_3$ is not a GRM but an ESOP because the same set of variables are used in more than one product term. We can write the following inclusion relationship for ESOP and the Reed-Muller forms; $PPRM \subseteq FPRM \subseteq GRM \subseteq ESOP$.

Due to the total freedom of input polarity and product term selection, the minimum number of product terms required to represent an arbitrary function in ESOP form can never be larger than the minimum number of product terms in any of the canonical Reed-Muller forms [3]. This fact can be seen from the arithmetic benchmark circuits given in Table 1, which was presented in [7]. Notice that PPRM yields the largest number of product terms since it is the most restricted form of AND-EXOR networks. In most cases, an ESOP realization gives a significantly smaller number of product terms even over the least restricted Reed-Muller form, GRM. This observation provides strong motivation for developing a testable ESOP implementation.

Another aspect of AND-EXOR representations presented in Table 1 is that AND-EXOR representations (especially ESOP) usually yield fewer product terms than a SOP representation. As we will illustrate later, this may be an area and delay advantage when realizing the function in 2-level form.

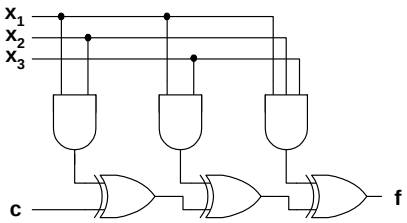
Our main contributions described in this paper are a highly testable ESOP realization and a minimal universal test set that detects all possible single stuck-at faults in the entire circuit, including the faults in the primary input and output leads. Another contribution of our work is a special built-in pattern gen-

- *U. Kalay, Marek A. Perkowski, Douglas V. Hall are with the Dept. of Electrical and Computer Engineering, Portland State University, Portland, OR 97207-0751.*
  E-mail: {ugurkal,mperkows,dough}@ee.pdx.edu.

For more information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number 107101.

Function	PPRM	FPRM	GRM	ESOP	SOP
adr4	34	34	34	31	75
log8	253	193	105	96	123
nrm4	216	185	96	69	120
rdm8	56	56	31	31	76
rot8	225	118	51	35	57
sym9	210	173	126	51	84
wgt8	107	107	107	58	255

Table 1: The number of product terms required to realize some arithmetic functions for different forms.



(a)

$T_1 = \begin{bmatrix} c & x_1 & x_2 & x_3 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$ $T_2 = \begin{bmatrix} c & x_1 & x_2 & x_3 \\ - & 0 & 1 & 1 \\ - & 1 & 0 & 1 \\ - & 1 & 1 & 0 \end{bmatrix}$

"-" : *don't care*

(b)

Figure 1: (a) A PPRM network implemented according to Reddy's scheme given in [2]. (b) Reddy's test set for the PPRM implementation.

erator, which gives 100% fault coverage for single stuck-at faults and has a much shorter testing cycle than a Pseudo-Exhaustive or Pseudo-Random Pattern Generator (PRPG). In addition, the hardware overhead for our special pattern generator is comparable to that of Linear Feedback Shift Register (LFSR) based pattern generators, such as PRPG.

The organization of this paper is as follows. In Section 2, some background on previous researchers' work is given. Section 3 describes our testable realization and the test set for it. In Section 4, we give a preliminary circuit, which can be used to generate our test set for BIST applications. In Section 5, we present our experimental results from area, delay, and test set size measurements performed on some benchmark circuits, along with the comparisons of our scheme with other schemes. Section 6 summarizes our results and gives some possible directions for future work.

2 BACKGROUND AND EARLIER WORK

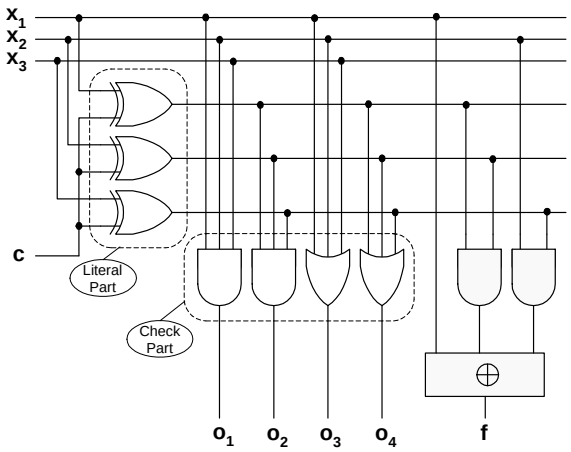
Reddy showed that a PPRM network can be tested for single stuck-at faults with a universal test set that is independent of the function being realized [2]. Figure 1a shows an EXOR cascade implementation of the PPRM expression $f = x_1x_2 \oplus x_1x_3 \oplus x_1x_2x_3$. In normal mode of operation, the control input, c, is set to the constant term in the functional expression (in this example: '0'). In testing mode, c is set according to the test set given in Figure 1b.

The four tests in test set T_1 detect all single stuck at faults in the EXOR cascade by applying all input combinations to every EXOR gate, independent of the number of EXOR gates in the cascade. The test vector $\langle 1111 \rangle$ in T_1 and the *walking-zero* test vectors in test set T_2 detect a single stuck-at fault in the AND part of the circuit. Since the number of vectors in T_1 is always equal to 4 and the number of vectors in T_2 is always equal to the number of input variables, n, the cardinality of Reddy's universal test set is (n+4).

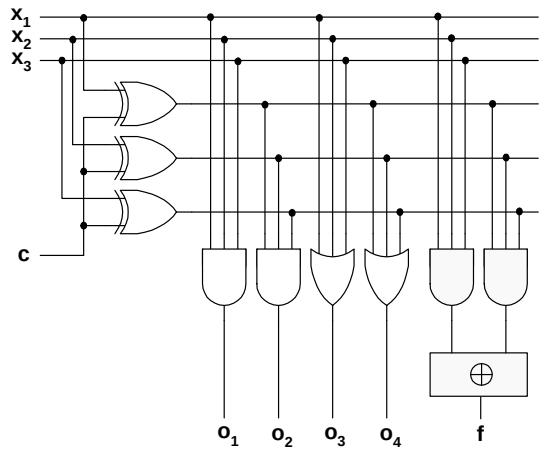
Reddy's technique is good for self testing because as shown by Daehn and Mucha [8], the entire test sequence can be inexpensively generated by a modified LFSR using a NOR gate and a shift register. However, as shown in Table 1, the number of terms in a PPRM is usually higher than the number of FPRM or GRM terms, and much higher than the number of ESOP terms [5]. For FPRM networks, Sarabi and Perkowski showed that by just inverting the test bits for the variables that are negative polarity in the FPRM, Reddy's test set can be used for single fault detection in a FPRM network [6]. They also showed that a GRM network can be decomposed into multiple FPRMs. This way, each FPRM can be tested separately to test the combined GRM circuit for single fault detection. The size of the test set, the worst case, is the number of FPRMs times (n+4). This method, however, does not yield a universal test set.

Other researchers have investigated multiple fault detection in AND-EXOR circuits. Sasao recently introduced a testable realization and a test set to detect multiple faults in GRM networks [7]. As shown in Figure 2a, Sasao uses an extra EXOR block, called the *Literal Part*, to obtain positive polarities for any negated variables and convert a GRM network into a PPRM network. When the control input c is set to '1', the shaded part of the circuit in Figure 2a realizes the GRM expression $f = x_1 \oplus \overline{x}_1\overline{x}_2 \oplus x_2\overline{x}_3$. The *Check Part* of the circuit in Figure 2a is added to test the literal part. Sasao implements the EXOR-Sum of the product terms with a tree structure instead of a cascade to obtain a less circuit delay. Nevertheless, his scheme does not lead to a universal test set. Furthermore, his scheme cannot be used for ESOP circuits because the conversion of an ESOP into PPRM by the literal part may produce an AND-EXOR expression that has multiple product terms with the same set of variables, which is not a PPRM form. For example, given the circuit in Figure 2b, the ESOP expression $f = x_1x_2x_3 \oplus \overline{x}_1\overline{x}_2\overline{x}_3$ yields two $x_1x_2x_3$ terms after the conversion from GRM into PPRM; one for $x_1x_2x_3$ and the other for $\overline{x}_1\overline{x}_2\overline{x}_3$.

Pradhan also targeted detection of multiple stuck-at faults in AND-EXOR circuits [3]. As shown in Figure 3, he does the negation of the literals by using an extra EXOR block called the *Control Block*. He uses cascaded AND gates in a *Check Block* to detect the faults in the control block. Figure 3 shows



(a)



(b)

Figure 2: (a) Sasao's GRM realization scheme, (b) Realizing an ESOP circuit with Sasao's scheme.

Pradhan's testable ESOP implementation for the function $f = x_1x_2x_3 \oplus \overline{x}_1\overline{x}_2\overline{x}_3$. Like Reddy, he implemented the EXOR-Sum of the product terms with a cascade structure.

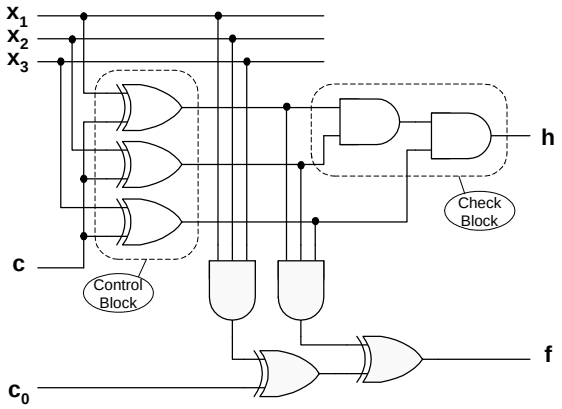


Figure 3: Pradhan's testable ESOP implementation.

Pradhan introduced a test set to detect all of the multiple faults in his testable realization for ESOP expressions. However, his test set is not universal and is too large to be practical for single fault detection. The cardinality of Pradhans test set is:

$$C = 6 + 2n + \sum_{e=0}^{j} \binom{n}{e},$$

where j is the order of the ESOP expression. The *order* is simply the maximum number of literals contained in any of the product terms. Notice that the complexity of the test is exponential with respect to the number of literals in product terms. Furthermore, if a product term has all possible literals, the test set is even larger than exhaustive (2^n) due to the additional test inputs required. For example, if there are four variables in an ESOP expression (n=4), and the order of the expression is 4 (j=4), the exponential term in the formula;

$$\sum_{e=0}^{4} \binom{4}{e} = \binom{4}{0} + \binom{4}{1} + \binom{4}{2} + \binom{4}{3} + \binom{4}{4} = 2^4 = 16,$$

which is exhaustive. The size of the entire test set then is,

$$C = 6 + 2n + \sum_{e=0}^{j} \binom{n}{e} = 6 + 2 \times 4 + 16 = 30.$$

3 TESTING SCHEME FOR ESOP NETWORKS

In this section, we introduce an improved testing scheme to detect single stuck-at faults not only in the internal lines of the circuit, but also in the primary inputs and outputs of the most general AND-EXOR circuits, ESOP.

3.1 Easily Testable ESOP Implementation

Figures 4 shows a new testable implementation for an ESOP expression. The circuit has up to two extra observable outputs, o_1 and o_2, and two additional control inputs, c_1 and c_2. The *Literal Part*, named after the similar part in Sasao's testable realization, is added to convert the ESOP circuit into a positive polarity AND-EXOR expression during testing. We do not refer to the expression after conversion as a PPRM because, as mentioned earlier, it may have some repeated product terms with the same set of variables. The positive polarity AND-EXOR expression cannot be tested by Sasao's multiple fault detection scheme [7], but can be tested by our single fault detection scheme. The *AND Part* and the *Linear Part* implement the desired ESOP expression as the c_1 control input is set to '1'. The f output is implemented with an EXOR cascade so that Reddy's universal tests for PPRM can be used for our realization. For the same reason, the *Check Part*, which is required to test the literal part, is implemented with an EXOR cascade. The gates marked with 'A' and 'B' are added for the detection of faults in the primary inputs, and the control input c_1. They are required based on the function being implemented. We will later describe the cases where gates 'A' and 'B' are required and how each section of our realization is tested when we explain our test set in the next sections.

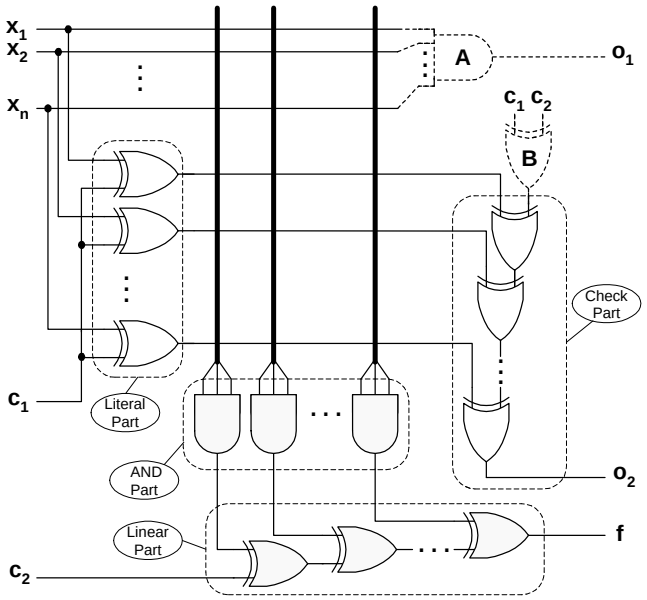


Figure 4: Easily testable ESOP circuit.

Hayes used mainly EXOR gates as additional circuitry to make a logic circuit easily testable [9]. Likewise, in our realization, we mainly use EXOR gates in the additional circuitry to take advantage of the superior testability properties of the EXOR gate. This allows us to obtain a minimal and universal test set.

3.2 The Fault Model

A fault model represents failures that affect functional behavior of logic circuits [10]. In a *stuck-at* fault model of a TTL AND gate, for example, an output could become shorted to V_{cc}. This can be modeled as a stuck-at 1 (s-a-1) fault. In MOS technology, most of the probable faults are opens and shorts, which can also be modeled with the appropriate s-a-0 or s-a-1 fault [11].

We follow the same approach taken by the previous researchers presented in Section 2, and assume a *single stuck-at fault* model, which allows only one stuck-at fault in the entire circuit. We also adopt their testing model for the detection of stuck-at faults in individual logic gates. An n-input AND gate requires (n + 1) test vectors to detect a single stuck-at fault in its inputs or output. The tests for a 3-input AND gate, for example, would be: $\{111, 011, 101, 110\}$. In this test set, $\langle 111 \rangle$ detects a s-a-0 fault on any of the inputs or the output, and the remaining tests, commonly referred to as *walking-zero* tests, detect a s-a-1 on any of the inputs and the output of the gate.

Some researchers (i.e. Pradhan [3], and Saluja, et al. [12]) analyzed in detail the fault characteristics of the EXOR gate implementation shown in Figure 5. They considered the faults in the internal lines of the EXOR gate as well as the faults in the inputs and the output of the EXOR gate. Table 2 shows the possible 16 functions that a 2-input circuit can implement. The circuit in Figure 5 realizes the function g_1, EXOR. If only a single stuck-at fault can occur in this implementation, the gate produces one of the 10 functions in Class A (g_{2-11}), and it can never produce the functions in Class B (g_{12-16}). Table 3 gives a mutually exclusive list of the faults in Class A as they are detected for each test applied to an EXOR gate.

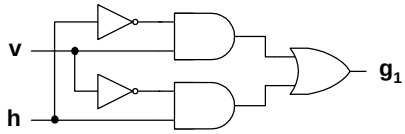


Figure 5: The EXOR implementation assumed by Pradhan and Reddy (et al.).

Inputs	$\oplus$	Class A										Class B				
v h	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}	g_{11}	g_{12}	g_{13}	g_{14}	g_{15}	g_{16}
0 0	0	0	0	0	1	1	1	0	0	1	0	0	1	1	1	1
0 1	1	0	0	0	1	1	1	1	1	0	1	0	1	0	0	0
1 0	1	0	1	1	0	1	1	0	1	1	0	0	0	0	1	0
1 1	0	0	0	1	0	0	1	0	1	0	1	1	1	0	1	1

Table 2: Functions that a 2-input logic gate can implement.

For our work we use the EXOR model in Figure 5, and the exhaustive test set in Table 3 to detect a single stuck-at fault in this model. By setting the c_1 and c_2 inputs in the proposed realization to the appropriate logic values, Reddy's four tests provide all input combinations for each EXOR gate in the EXOR cascades (the linear part and the check part) of our realization.

v	h	Faults Detected
0	0	g_5, g_6, g_7, g_{10}
0	1	g_2, g_3, g_4
1	0	g_8, g_{11}
1	1	g_9

Table 3: The faults exclusively detected by all of the input vectors applied to an EXOR.

3.3 The Test Set

3.3.1 Fault detection in the internal lines of the realization

The linear part of the proposed ESOP circuit can be tested by Reddy's T_1 test set. During the testing of this part, the AND gate inputs are either all 0's or all 1's. This makes the AND part transparent to the linear part because all-0's or all-1's are transferred to the external inputs of the EXOR cascade in the linear part. The network response to the test vectors is observed from the function output f. The test set for the linear part, T_a, is given below.

$$T_a = \begin{array}{c} \begin{array}{ccccccccc} c_1 & c_2 & x_1 & x_2 & x_3 & \cdots & x_n \end{array} \\ \left[\begin{array}{ccccccc} 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 1 & 1 & \cdots & 1 \\ 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 1 & 1 & 1 & \cdots & 1 \end{array}\right] \end{array}$$

During the testing of the linear part, the check part of the circuit receives the same test vectors given in T_a, and is therefore tested at the same time. However, for the check part, output o_2 is observed instead of output f for the response to the test vectors.

The AND part of the circuit is tested for single stuck-at faults in the same way as in Reddy's scheme. The test vectors are applied to the primary inputs and transferred to the AND part by setting c_1 control input to '0'. The test set T_2 is applied to detect a s-a-1 in any input and the output of the AND gates. The complete test set for the AND gates, T_b, is obtained as below by including the test vector $\langle$0-11$\cdots$1$\rangle$ to detect a s-a-0 in any of the inputs and the outputs of the AND gates. A '-' denotes a don't care logic value.

$$T_b = \begin{array}{c} \begin{array}{ccccccccc} c_1 & c_2 & x_1 & x_2 & x_3 & \cdots & x_n \end{array} \\ \left[\begin{array}{ccccccc} 0 & - & 0 & 1 & 1 & \cdots & 1 \\ 0 & - & 1 & 0 & 1 & \cdots & 1 \\ & & & & \vdots & & \\ 0 & - & 1 & 1 & 1 & \cdots & 0 \\ \hline 0 & - & 1 & 1 & 1 & \cdots & 1 \end{array}\right] \end{array}$$

The literal part is tested through the check part and the extra observation output, o_2. Again, in the case of a fault, any logic change in the output of the EXOR gates in the literal part is propagated to the observable output o_2. The four tests given in T_c apply all input combinations to each EXOR gate in the literal part.

$$T_c = \begin{array}{c} \begin{array}{ccccccccc} c_1 & c_2 & x_1 & x_2 & x_3 & \cdots & x_n \end{array} \\ \left[\begin{array}{ccccccc} 0 & - & 0 & 0 & 0 & \cdots & 0 \\ 0 & - & 1 & 1 & 1 & \cdots & 1 \\ 1 & - & 0 & 0 & 0 & \cdots & 0 \\ 1 & - & 1 & 1 & 1 & \cdots & 1 \end{array}\right] \end{array}$$

3.3.2 Fault detection in the circuit input/output leads

The primary inputs that are applied to the literal part are tested through the path formed by the literal part, check part, and the observable output o_2. The required test set of this case, T_d, is given below. The first vector detects a s-a-1 fault, and the second vector detects a s-a-0 fault.

$$T_d = \begin{array}{c} \begin{array}{ccccccccc} c_1 & c_2 & x_1 & x_2 & x_3 & \cdots & x_n \end{array} \\ \left[\begin{array}{ccccccc} 0 & - & 0 & 0 & 0 & \cdots & 0 \\ 0 & - & 1 & 1 & 1 & \cdots & 1 \end{array}\right] \end{array}$$

The primary inputs that are not applied to the literal part (when used only in positive polarity in the expression), but that are applied an odd number of times to the AND part are tested through the path formed by the AND part, the linear part, and the function output f. The required test set, T_e, is given below. Any stuck-at fault in the primary inputs of this class causes an odd number of changes in the external inputs of the linear part, and is detected by observing the function output f.

$$T_e = \begin{array}{c} \begin{array}{ccccccccc} c_1 & c_2 & x_1 & x_2 & x_3 & \cdots & x_n \end{array} \\ \left[\begin{array}{ccccccc} - & - & 0 & 0 & 0 & \cdots & 0 \\ - & - & 1 & 1 & 1 & \cdots & 1 \end{array}\right] \end{array}$$

The primary inputs that are not applied to the literal part, and are applied an even number of times to the AND part cannot be tested with the above test set because an even number of value changes cannot be propagated to the output by the EXOR cascade in the linear part. Therefore, the additional AND gate 'A' with the observable output o_1 is required for the primary inputs of this class. The same scheme is described by Reddy in [2]. However, the chance of having this extra AND gate is less likely in our scheme because of the alternative path from the primary inputs to the observable output o_2 when the primary inputs are applied to the literal part. The required test vectors for the primary inputs of this class and for the observable output o_1 are covered by the required test vectors to test the extra AND gate 'A', which are given in T_f. Notice that the faults in the primary inputs and the faults in the AND gate inputs are equivalent; and similarly, the faults in the observable output o_1 and the faults in the AND gate output are equivalent.

$$T_f = \begin{array}{c} \begin{array}{ccccccccc} c_1 & c_2 & x_1 & x_2 & x_3 & \cdots & x_n \end{array} \\ \left[\begin{array}{ccccccc} - & - & 0 & 1 & 1 & \cdots & 1 \\ - & - & 1 & 0 & 1 & \cdots & 1 \\ & & & & \vdots & & \\ - & - & 1 & 1 & 1 & \cdots & 0 \\ \hline - & - & 1 & 1 & 1 & \cdots & 1 \end{array}\right] \end{array}$$

The faults in the control input c_1 are detected through the path formed by the literal part, check part, and the observable output o_2. Detection through the path to the function output f cannot be guaranteed because it is dependent on the function being implemented. If the number of the literal part outputs (the number of EXOR gates in the literal part) is an odd number, the extra EXOR gate 'B' is not required and c_2 is by-passed to the output of this extra gate. Note that all of the literal part outputs will change at the same time in case of a fault in c_1. Therefore, if the number of changes at the output of the literal part is an odd number, the EXOR cascade in the check part will propagate the fault to o_2. The required test set, T_g, is given below. The first vector detects a s-a-1, and the second vector detects a s-a-0 in c_1.

$$T_g = \begin{array}{c} \begin{array}{ccccccccc} c_1 & c_2 & x_1 & x_2 & x_3 & \cdots & x_n \end{array} \\ \left[\begin{array}{ccccccc} 0 & - & - & - & - & \cdots & - \\ 1 & - & - & - & - & \cdots & - \end{array}\right] \end{array}$$

If the number of the literal part outputs is an even number, then the extra EXOR gate 'B' is required to make the number of changes fed into the check part an odd number. This configuration also allows the use of the same test set, T_g, above for the detection of faults in c_1. However, the extra EXOR gate needs to be tested, as well. The test set for this EXOR gate, T_h, is exhaustive by its testing model, and given below.

$$T_h = \begin{array}{c} \begin{array}{ccccccccc} c_1 & c_2 & x_1 & x_2 & x_3 & \cdots & x_n \end{array} \\ \left[\begin{array}{ccccccc} 0 & 0 & - & - & - & \cdots & - \\ 0 & 1 & - & - & - & \cdots & - \\ 1 & 0 & - & - & - & \cdots & - \\ 1 & 1 & - & - & - & \cdots & - \end{array}\right] \end{array}$$

The faults in primary output f and the control input c_2 are covered by the test set T_a of the linear part due to fault equiv-

alence. Similarly, the faults in the observable output o_2 are covered by the test set T_a of the check part.

3.3.3 The complete test set

Theorem: An ESOP circuit with the realization in Figure 4 can be tested for single stuck-at faults in its internal lines and in its input/output leads requiring a test set of (n + 6) cardinality.

Proof: A test set, T, that covers all of the test sets above, T_{a-h}, detects a single stuck-at fault in the entire circuit. The minimal test set then is: $T = T_a \cup T_b \cup T_c \cup T_d \cup T_e \cup T_f \cup T_g \cup T_h$.

The cardinality of the minimal test set is obtained as follows.

- include T_a in T ..... (4 tests),
- combine* the last two vectors of T_c and T_h and include in T .....(2 tests),
- include the first n vectors of T_b in T ..... (n tests).

Total: (n + 6)

The remaining tests are covered by T as follows:

- the last vector of T_b, the first two vectors of T_c, the test set T_d, the test set T_e, the last vector of T_f, the first vector of T_g, the first vector of T_h are covered by the first two vectors of T_a, which are included in T.
- the first n vectors of T_f are covered by the first n vectors of T_b, which are included in T.
- the second vector of T_g is covered by the last vector of T_c, which is included in T.
- the second vector of T_h is covered by the third vector of T_a, which is included in T.

The final test set T:

$$T = \begin{array}{c} \begin{array}{ccccccc} c_1 & c_2 & x_1 & x_2 & x_3 & \cdots & x_n \end{array} \\ \left[\begin{array}{ccccccc} 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 1 & 1 & \cdots & 1 \\ 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 1 & 1 & 1 & \cdots & 1 \\ \hline 0 & - & 0 & 1 & 1 & \cdots & 1 \\ 0 & - & 1 & 0 & 1 & \cdots & 1 \\ & & & \vdots & & & \\ 0 & - & 1 & 1 & 1 & \cdots & 0 \end{array}\right] \end{array}$$

□

*The combining process over two test vectors is done by replacing the don't care values of the first test vector with the determined values of the second test vector.

Although we did not prove that there is not a shorter universal test set than (n + 6) for general ESOP, this result is very close to the lower bound of the length of a universal test set, (n + 4), for 2-level AND-EXOR networks [2, 13, 14]. Note that by modifying Reddy's test set based on the function being realized, Kodandapani introduced a test set with (n+3) tests [15], but his test set is not universal.

3.4 Example

Figure 6 shows our testable realization for the ESOP expression $f = x_1x_5 \oplus x_1x_2x_3 \oplus x_2x_3x_4 \oplus \overline{x}_2\overline{x}_3\overline{x}_4$. In this example, the extra observable output o_1 is required for the detection of the faults in primary input x_1 since x_1 is not applied to the literal part and it is used an even number of times in the AND part. The primary input x_5 is not applied to the literal part, either, but it is used an odd number of times in the AND part. Note that the extra AND gate 'A' is not required since there is only one primary input to observe at o_1. The extra EXOR gate 'B' is also not required because the number of EXOR gates in the literal part is an odd number.

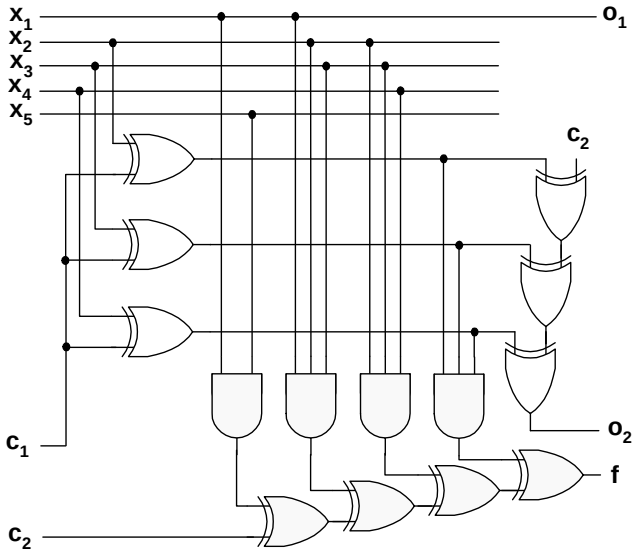


Figure 6: An example testable ESOP realization.

The test set for the example implementation is:

$$T = \begin{array}{c} \begin{array}{ccccccc} c_1 & c_2 & x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \left[\begin{array}{ccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 0 & - & 0 & 1 & 1 & 1 & 1 \\ 0 & - & 1 & 0 & 1 & 1 & 1 \\ 0 & - & 1 & 1 & 0 & 1 & 1 \\ 0 & - & 1 & 1 & 1 & 0 & 1 \\ 0 & - & 1 & 1 & 1 & 1 & 0 \end{array}\right] \end{array}$$

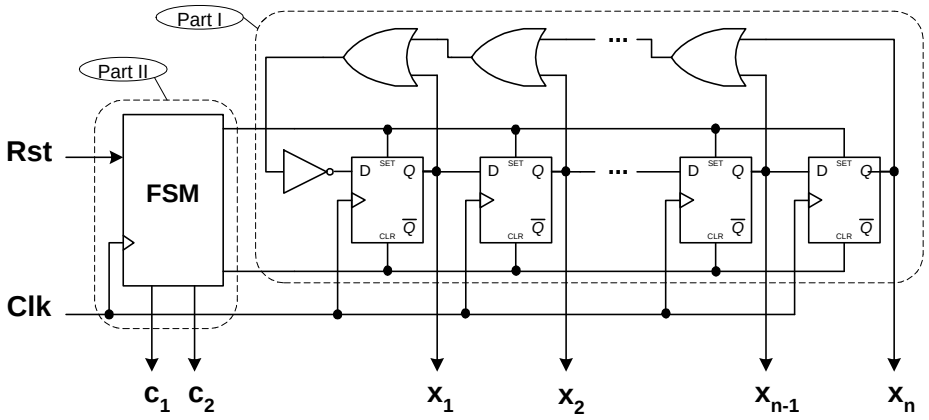


Figure 7: An example EDPG circuit implementation.

4 BUILT-IN SELF TEST CIRCUITRY FOR ESOP CIRCUITS

A traditional, signature analysis based Built-in Self Test (BIST) circuitry for a combinational network consists mainly of a pattern generator, and a signature register. For the complete testing system, a BIST controller, some multiplexers, a comparator, and a ROM are also embedded inside the chip. Using a Linear Feedback Shift Register (LFSR)-based Pseudo-Exhaustive or Pseudo-Random Pattern Generator (PRPG) is a well-known method for generating test patterns for very large and complex combinational circuits. The PRPG approach is used because it is difficult to otherwise generate the large and irregular test sets required by such combinational circuits. As shown by Drechsler (et al.) in [16], the PRPG approach does not work better with AND-EXOR circuits than it does with equivalent SOP circuits. However, they show that AND-EXOR networks do have good deterministic testability performance. Considering this fact and the properties of our design, we can list the reasons why we propose a deterministic test generation for built-in self test of ESOP circuits as below;

1. Our ESOP realization is designed for testing and therefore requires a minimal test set. Traditional PRPG test length is much longer than our test set for the same fault coverage. Also, there is no need for partitioning the circuit to prevent the long cycles of a pseudo-exhaustive test generation.

2. Our test set is universal, which allows it to be generated by fixed hardware that can be used for any function.

3. Our test set has regular patterns and therefore easy to generate. As a result, the area overhead of our pattern generator in our scheme is comparable to that of PRPG based schemes.

4. Our test set gives 100% fault coverage for single fault detection and does not require fault simulation.

Figure 8 shows the BIST circuitry for ESOP circuits. In this structure, the only difference from classical BIST circuitry is the *ESOP Deterministic Pattern Generator* (EDPG) that is introduced for our easily testable ESOP implementation. The results from the applied test vectors are collected from the function output f and from the extra observable outputs o_1 and o_2; then compressed in the signature register, which can simply be an LFSR based Multiple Input Signature Register (MISR) [1]. After all the tests are applied, the signature register content is compared with the correct signature of the implemented ESOP to generate a *go/no go* signal at the end of the test cycle.

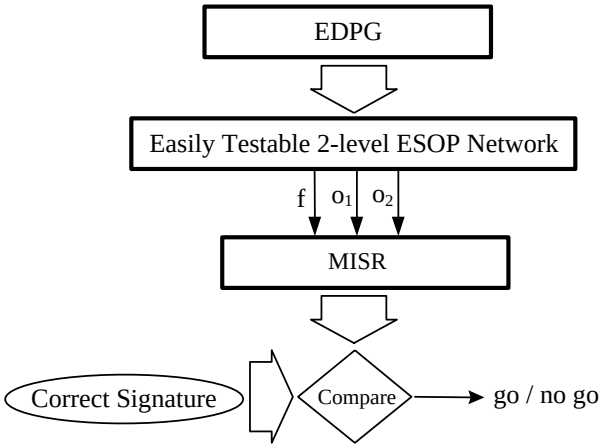


Figure 8: The BIST circuitry for highly testable ESOP circuits.

A real life circuit is more likely to have multiple outputs rather than a single output as shown in the earlier examples. In this case, the AND gates in the AND part (the product terms) are distributed over multiple linear parts (EXOR cascades) for multiple outputs, and therefore the faults must be observed from all circuit outputs. Also, for a multi-output circuit, all the function outputs should be applied to the MISR along with the extra observable outputs.

Daehn and Mucha designed a simple BIST circuit to test PLAs [8]. They used LFSRs and NOR gates to generate regular test patterns such as a *walking-one* test sequence. Similarly, our EDPG can be built to generate the *walking-zero* sequence along with the extra c_1 and c_2 bits, as shown in Figure 7.

Part I of EDPG generates the walking-zero portion of the test vectors. This portion of the BIST circuitry can be expanded linearly based on the number of inputs in the ESOP circuit. Part

II of the EDPG is a Finite State Machine (FSM), and generates c_1 and c_2 bits of the test vectors. It also provides $\overline{\text{CLR}}$ and $\overline{\text{SET}}$ signals for the D-Flip-flops in Part I to generate all-0 or all-1 bits in the first six test vectors of the test set, T. Part II is independent of the function being realized, and therefore is fixed size. Figure 9 gives the state diagram and the circuit implementation for the FSM in Part II. The FSM generates the six vectors of the test set, then stops and enables Part I to generate the walking-zero tests. Figure 10 gives the simulation results for the EDPG implementation.

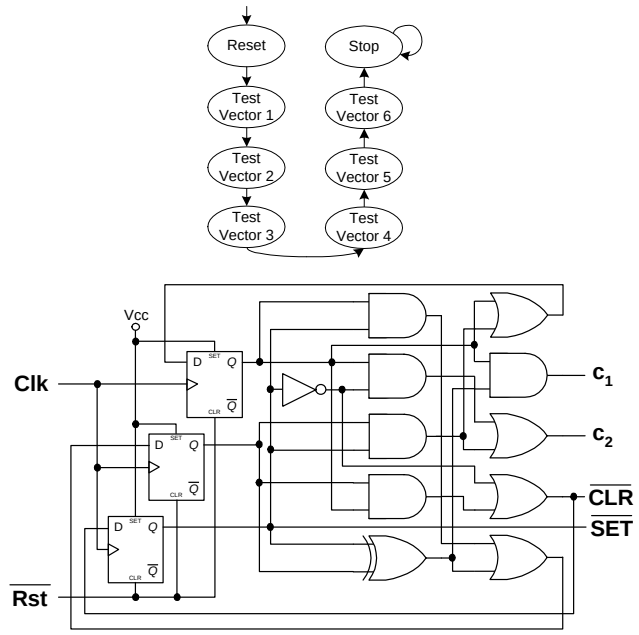


Figure 9: State diagram and the circuit implementation for Part II of EDPG.

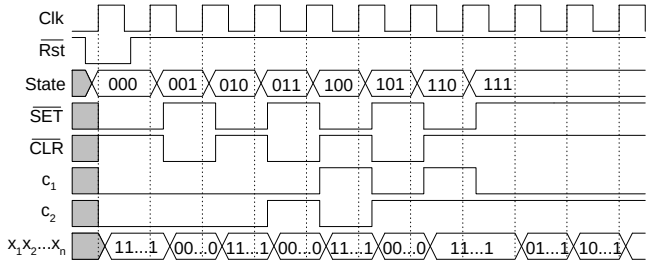


Figure 10: Simulation of the EDPG circuit.

5 EXPERIMENTAL RESULTS

We performed area, delay, and test set size measurements on some benchmark circuits using our realization scheme and 2-level/multi-level synthesis schemes. We selected the circuits from LGSynth'93 and Espresso benchmark sets to provide a wide variety of function types. For example, we selected circuits with different numbers of primary inputs; implementations in 2-level or multi-level; and of different classes such as math, logic, etc.

All the circuits were optimized for area and mapped to a technology library before performing the measurements. SIS [17] was used to optimize and synthesize the circuits in multi-level; and Exorcism [18] was used to optimize ESOP expressions. For a multi-level benchmark circuit, we used SIS to obtain the equivalent two-level SOP expression; and used Disjoint [19] to convert it to a two-level AND-EXOR expression before optimizing with Exorcism.

A 0.5 micron, array-based library developed by LSI Logic Corp. [20] was used for synthesis. We limited the number of components in the library according to Table 4. All area measurements are expressed in cell units, excluding the interconnection wires.

Component	Function	Area (cells)	Delay(ns)	
			Block	Fanout
INV	O=!a	1	0.03	0.036
NAND2	O=!(a*b)	1	0.06	0.036
NAND3	O=!(a*b*c)	2	0.09	0.04
NAND4	O=!(a*b*c*d)	2	0.12	0.05
NOR2	O=!(a+b)	1	0.06	0.075
NOR3	O=!(a+b+c)	2	0.15	0.115
NOR4	O=!(a+b+c+d)	4	0.22	0.155
AND2	O=a*b	2	0.15	0.035
AND3	O=a*b*c	2	0.21	0.035
OR2	O=a+b	2	0.19	0.025
OR3	O=a+b+c	2	0.33	0.025
XOR2	O=a*!b+!a*b	3	0.3	0.035
XNOR2	O=a*b+!a*!b	3	0.28	0.04
AOI21	O=!(a*!b+c)	2	0.14	0.025
AOI22	O=!(a*!b+c*d)	2	0.18	0.075
OAI21	O=!((a+b)*c)	2	0.1	0.06
OAI22	O=!((a+b)*(c+d))	2	0.15	0.075

Table 4: The technology library used in measurements.

Table 5 gives the number of test vectors and the fault coverage obtained from different schemes for single faults. This table compares our test scheme with LFSR based pseudo-random test generation and with algorithmic test generation. Pseudo-random and algorithmic test vectors were generated for the multi-level implementation of the circuits, where our test set was generated (predetermined) for the easily testable ESOP implementation. SIS was used for algorithmic test generation. Up to 10,000 LFSR patterns were generated for each circuit using the program used in [21]. The goal was to cover a wide range of circuits to determine circuits that are random vector resistant and require more than 10,000 patterns. Almost half of the selected benchmark circuits required more than 10K pseudo-random patterns for 100% fault coverage, whereas our scheme required no more than 150 patterns for all the circuits. For example, the fault coverage for the circuit x9dn is 73.1% for 10K random patterns. In comparison, our scheme yields only 33 patterns for 100% fault coverage. In all circuits, our test set is smaller than either pseudo-random or algorithmic test sets. Also, note that algorithmic test sets are generally not universal and therefore cannot be utilized in a simple self-test circuit.

The next measurement was performed to see the testability

Circuit	Number of Primary Inputs	Multi-level Implementation						Our ESOP Implementation	
		Pseudo-random Test Set (LFSR)				Algorithmic Test Set (SIS)		Our Test Set	
		#Tests	Undetected	Total Faults	Fault Coverage	#Tests	Fault Coverage	#Tests	Fault Coverage
9symml	9	512	0	513	100	137	100	15	100
adr4	8	96	0	146	100	37	100	14	100
alu1	12	64	0	91	100	35	100	18	100
alu2	10	864	0	664	100	117	100	16	100
alu4	14	10K	108	4158	97.4	961	100	20	100
apex5	117	10K	979	4129	74.6	1245	100	121	100
apex6	135	10K	27	1680	98.3	400	100	141	100
ex4	128	10K	92	1042	91.1	474	100	134	100
f51ml	8	256	0	465	100	101	100	14	100
mux	21	320	0	104	100	53	100	27	100
rd73	7	128	0	341	100	78	100	13	100
x1	51	10K	50	1342	96.2	301	100	57	100
x2	10	320	0	105	100	26	100	16	100
x4	94	3744	0	991	100	280	100	100	100
x2dn	82	10K	48	467	89.7	147	100	88	100
x9dn	27	10K	129	480	73.1	126	100	33	100

Table 5: Comparisons of the number of test vectors for different circuits.

improvement of the proposed ESOP implementation that requires some additional gates and input/output pins (labeled in Table 6 as: "ESOP with DFT (Design for Test)"), over the ordinary 2-level ESOP implementation that does not include any additional hardware (labeled in Table 6 as: "Ordinary ESOP"). The test vectors for the ordinary ESOP implementation were algorithmically generated with the program used in [22], and compared with the universal test set of our implementation scheme. In Table 6, our test set is significantly smaller than the algorithmically generated test sets for the majority of the benchmark circuits. Only one of the benchmark circuits, alu1, yielded fewer number of algorithmically generated test patterns than our universal test set. However, as mentioned earlier, our test set is universal, which eliminates the need for test generation programs; and it has regular patterns, which can be generated easily. As a result, it is very suitable for BIST.

We did not generate pseudo-random vectors for our implementation scheme as an alternative to our EDPG for three reasons. First, the fact that AND-EXOR circuits are not more testable with pseudo-random patterns than AND-OR circuits was shown by Drechsler (et al.) in [16]; second, our ESOP implementation is constructed considering certain regular and minimal patterns, and therefore it requires those patterns for guaranteed 100% fault coverage; and third, as shown next, the area overhead for our EDPG is very close to that of LFSR based PRPGs.

In Table 8, the area of different patterns generators is given in cell units based on the library components given in Table 4. The total BIST area is not calculated for comparisons because, as mentioned earlier, the only difference between a classical BIST circuitry and the ESOP BIST circuitry is the pattern generators used in them. We selected circuits with a wide range of number of primary inputs since the area of a pattern generator

Circuit	Number of Primary Inputs	Algorithmic / Ordinary ESOP	Our Scheme / ESOP with DFT
9symml	9	181	15
a04	9	173	15
alu1	12	8	18
alu2	10	133	16
alu4	14	1174	20
dk16	7	77	13
mux	21	48	27
rd73	7	48	13
rd84	8	62	14
sse	11	60	17
x2	10	27	16

Table 6: Number of deterministic test vectors for two ESOP implementations.

is directly related to the number of primary inputs. An LFSR based pseudo-exhaustive or pseudo-random pattern generator mainly consists of D-Flip-flops and EXOR gates [23, 24]. The number of 2-input EXOR gates changes typically from one to the number of primary inputs (n), based on the characteristic polynomial used to generate the patterns. Therefore, the area of an LFSR-based PRPG is given as a range in the table. The area of a BILBO register is also included in the table since it is used for pseudo-random pattern generation [25].

As shown in Table 8, the area of EDPG is comparable to those of other pattern generators for all benchmark circuits. It is always better than BILBO's if $n \geq 8$, and is in the range of PRPG's if $n \geq 48$. For example, for the circuit rd73, the BILBO register is smaller than EDPG, but 128 pseudo-random

Circuit	Number of Primary Inputs	Number of Primary Outputs	Multi-level				2-level						
			Using All Components		Using AND2, OR2, INV		SOP			Our ESOP			
			Area	Delay	Area	Delay	#Terms	Area	Delay	#Terms	Area (Function)	Area (DFT)	Delay
9symml	9	1	193	2.9	468	4.13	87	617	2.33	51	530	27	18.15
adr4	8	5	54	1.38	123	2.16	75	362	1.98	31	233	27	5.85
alu1	12	8	31	0.6	74	0.89	19	44	0.67	16	114	33	1.58
alu2	10	6	320	7.72	685	10.28	260	1998	2.58	69	643	33	13.8
alu4	14	8	1800	11.38	6020	14.75	1138	10278	2.93	455	5603	45	69.74
apex5	117	88	1482	5.73	3517	7.46	1216	6901	2.22	399	4951	185	5.09
apex6	135	99	694	2.39	1537	3.57	657	3548	2.22	408	4127	221	13.08
ex4	128	28	439	2.11	1029	3.51	559	3925	2.33	317	3829	299	14.47
f51ml	8	8	106	6.62	256	8.86	97	522	1.98	31	238	21	4.51
mux	21	1	36	1.76	96	2.43	16	116	1.62	16	156	15	6.18
rd73	7	3	118	4.04	279	5.07	187	1187	2.33	41	304	21	9.87
x1	51	35	307	2.39	669	3.19	324	2124	2.58	414	5834	133	32.56
x2	10	7	50	1.11	111	1.88	31	76	1.27	15	144	27	2.83
x4	94	71	456	5.18	921	3.65	531	2407	1.87	297	2906	243	5.42
x2dn	82	56	182	1.23	375	2.43	120	477	1.87	101	865	155	5.76
x9dn	27	7	144	1.99	320	3.03	120	1141	2.22	184	2605	77	25.86

Table 7: Area and delay comparisons for different implementation schemes.

Circuit	Num. of Primary Inputs	Area of PRPG (LFSR)		Area of BILBO	Area of EDPG
		1 EXOR	*n* EXORs		
alu4	14	129	168	236	202
apex1	45	408	540	747	543
apex5	117	1056	1404	1935	1335
i2	201	1812	2412	3321	2259
rd73	7	66	84	120	125
x4	94	849	1128	1556	1082
x9dn	27	246	324	450	345

Table 8: Area measurements in cell units for different pattern generators.

patterns are required for 100% fault coverage, where only 13 EDPG patterns are required for the same fault coverage. Another advantage of EDPG is that it does not need an initialization seed, unlike most of the LFSR based pattern generators that require one or more seeds. We did not include the area overhead of the additional hardware that provides the initialization for the PRPGs in Table 8.

Table 7 presents the results of another set of measurements to show the area and delay performance of our ESOP implementation in Figure 4 as compared to the multi-level and 2-level SOP implementations. The area information is separated into two parts for the ESOP implementation. Those are, the area required for the implementation of the function being implemented (the literal part, the AND part, and the linear part), and the area required for the gates added for better testability (the gates 'A' and 'B', and the check part), denoted in the table as "Design for Testing" (DFT).

The measurements on multi-level implementations are performed for two different cases: one by using the entire set of library components presented in Table 4, and the other by using only AND2, OR2, and INV gates. These two different measurements were performed to see the variations in area and delay as the components in the targeted library were changed for synthesis. This provides a more objective comparison.

A column is provided for 2-level SOP implementations to evaluate our design in the PLA environment. The delay information for a 2-level SOP circuit is calculated by assuming a tree-of-OR-gates structure (using 3-input and 2-input OR gates) to combine the product terms. Similarly, the AND gates with more than three inputs in both SOP and our ESOP implementations were implemented as a tree of smaller AND gates (3-input and 2-input). The tree structure assumption does not affect the testing of the AND gates in our ESOP scheme. Another comparison of 2-level SOPs and ESOPs is given by Saul (et al.) in [26] for PLA and XPLA implementations.

Although it is not fair to compare a cascade implementation to a tree-like implementation, Table 7 shows that our 2-level ESOP implementation is comparable to multi-level implementations in most of the cases. For example, the ESOP implementations of apex5 and f51ml have better delay than multi-level implementations. Also, adr4, alu1, mux, x2, x4, and x2dn have fairly low delays when implemented with our ESOP scheme. In a few cases, our ESOP scheme yielded significantly larger delays than multi-level and 2-level SOP implementations, such as for the circuits alu4, x1, and x9dn.

Similarly, the ESOP circuits implemented for alu2, alu4, and f51ml have areas between the area of their multi-level version implemented using all library gates and the area of their multi-level version implemented using only AND2, OR2, and INV gates. Also the areas of 9symml, alu1, and rd73 are very close to those of their multi-level versions.

As 2-level implementation comparisons, for 50% of the

benchmark circuits, our ESOP implementation scheme yielded smaller area than 2-level SOP implementation. Especially for circuits rd73,alu2,f51ml, and alu4 the areas of SOP implementations are 3.65, 3, 2.01, and 1.8 times larger, respectively, than those of ESOP implementations.

The area overhead for DFT in our ESOP implementation is typically less than 0.1% (the least, 0.001%, for alu4). And, the largest area overhead, 28%, was obtained for alu1 since the functionality of the circuit is relatively small in comparison to those of the other benchmark circuits.

6 CONCLUSIONS AND FUTURE RESEARCH

In this paper, we have shown a highly testable ESOP realization and a minimal universal test set for the detection of single stuck-at faults in both internal lines and primary inputs/outputs of the circuit with 100% fault coverage. An EXOR cascade is used in the check part instead of AND gates or OR gates because the EXOR cascade yields much fewer test vectors and today's advanced technology makes it possible to have an EXOR gate almost as fast as an AND gate.

The experimental results show that our test set is always smaller in exponential degree than pseudo-random test set, and smaller in multiples than an algorithmically generated test set for 100% single stuck-at fault coverage. A deterministic test pattern generator is presented to be used as a part of the built-in self-test circuitry. The experimental results show that the overall overhead of our BIST circuit is comparable to that of the traditional PRPG based BIST circuit. More importantly, our pattern generator is superior to a PRPG because of its 100% single fault coverage and much shorter testing cycle. Also, it does not require an initialization seed and the circuitry for generating it. The results also show that our 2-level ESOP implementation is comparable to (or in some cases better than) the multi-level and 2-level SOP implementations in the area and delay measurements. Furthermore, our implementation gives a very small DFT area overhead.

In addition to detecting all single stuck-at faults, our architecture and test set detect a significant fraction of multiple stuck-at faults. More tests can be added to improve the multiple fault coverage even though it is very unlikely that a minimal and universal test set can be found. For instance, more zero-weighted test vectors can be added to improve the multiple fault coverage for the AND part of the circuit as explained by Saluja and Reddy in [12]. Another method to improve the fault coverage is to detect multiple faults with the help of multiple outputs in a multi-output ESOP circuit. Our BIST methodology with an MISR is an ideal method for this purpose. Our test set can also be improved for bridging faults using a method similar to that presented by Bhattacharya (et al.) in [27]. We are currently investigating our test set and implementation scheme for detecting bridging faults and multiple faults. The results will be presented in our next paper.

Acknowledgments

The authors would like to thank Prof. W. Robert Daasch at Portland State University for helping with area calculations; Craig M. Files at Portland State University for a review of the paper; Prof. Nur A. Touba at Univ. of Texas at Austin for providing the program used in [21]; and Prof. L. Jozwiak and A. Slusarczyk at Eindhoven University for providing the program used in [22] for our measurements.

References

[1] M. Abramovici, M. A. Breuer, and A. D. Friedman, *Digital Systems Testing and Testable Design*. New York: IEEE Press, 1990.

[2] S. M. Reddy, "Easily Testable Realizations for Logic Functions," *IEEE Trans. on Comp.*, vol. C-21, pp. 1183–1188, Nov. 1972.

[3] D. K. Pradhan, "Universal Test Sets for Multiple Fault Detection in AND-EXOR Arrays," *IEEE Trans. on Comp.*, vol. 27, pp. 181–187, Feb. 1978.

[4] H. Fujiwara, *Logic Testing and Design for Testability*. MIT Press, 1985.

[5] T. Sasao, *Logic Synthesis and Optimization*. Norwell, MA: Kluwer Academic Publishers, 1993.

[6] A. Sarabi and M. A. Perkowski, "Design for Testability Properties of AND-EXOR Networks," in *Int. Workshop on Applications of the Reed-Muller Expansion in Circuit Design*, pp. 418–424, Sept. 1993.

[7] T. Sasao, "Easily Testable Realizations for Generalized Reed-Muller Expressions," *IEEE Trans. on Comp.*, vol. 46, pp. 709–716, June 1997.

[8] W. Daehn and J. Mucha, "A Hardware Approach to Self-Testing of Large Programmable Logic Arrays," *IEEE Trans. on Comp.*, vol. C-30, pp. 829–833, Nov. 1981.

[9] J. P. Hayes, "On modifying logic networks to improve their diagnosability," *IEEE Trans. on Comp.*, vol. C-23, no. 1, pp. 56–62, 1974.

[10] F. Lombardi and M. Sami, *Testing and Diagnosis of VLSI and ULSI*, ch. Fault Detection and Design for Testability of CMOS Logic Circuits (by S. M. Reddy and S. Kundu), pp. 69–91. Boston: Kluwer Academic Publishers, 1988.

[11] J. M. Cortner, *Digital Test Engineering*. New York: John Wiley & Sons, 1987.

[12] K. K. Saluja and S. M. Reddy, "Fault Detecting Test Sets for Reed-Muller Canonic Networks," *IEEE Trans. on Comp.*, vol. C-24, pp. 995–998, Oct. 1975.

[13] T. R. Damarla and M. Karpowsky, "Fault Detection in Combinational Networks by Reed-Muller Transforms," *IEEE Trans. on Comp.*, vol. 38, pp. 788–797, June 1989.

[14] H. Fujiwara, "On Closedness and Test Complexity of Logic Circuits," *IEEE Trans. on Comp.*, vol. 30, pp. 556–562, Aug. 1981.

[15] K. L. Kodandapani, "A Note on Easily Testable Realizations for Logic Functions," *IEEE Trans. on Comp.*, vol. C-23, pp. 332–333, 1974.

[16] R. Drechsler, H. Hengster, H. Schfer, J. Hartmann, and B. Becker, "Testability of 2-level AND/EXOR Circuits," in *European Design & Test Conf.*, (Paris, France), Mar. 1997.

[17] E. M. Sentovich, J. K. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. R. Stephan, R. K. Brayton, and A. Sangiovanni-Vincentelli, "SIS: A System for Sequential Circuit Synthesis," Tech. Rep. UCB/ERL M92/41, Electronics Research Laboratory, UCLA, Berkeley, CA, May 1992.

[18] N. Song and M. Perkowski, "Minimization of Exclusive Sum of Products Expressions for Multi-Output Multiple-Valued Input, Incompletely Specified Functions," *IEEE Trans. on Computer Aided Design*, vol. 15, pp. 285–395, Apr. 1996.

[19] B. J. Falkowski and M. A. Perkowski, "An Algorithm for the Generation of Disjoint Cubes for Completely and Incompletely Specified Boolean Functions," *Int. J. Electronics*, vol. 70, no. 3, pp. 533–538, 1991.

[20] LSI Logic Corporation, *LCA/LEA500K Array-Based Products Databook*, fourth ed., May 1997.

[21] Z. Zhao, B. Pouya, and N. A. Touba, "BETSY: Synthesizing Circuits for a Specified BIST Environment," in *Proc. of Int. Test Conf.*, pp. 144–153, 1998.

[22] A. Slusarczyk, L. Jozwiak, and M. A. Perkowski, "Term Trees in Application to an Effective and Efficient ATPG for AND-EXOR and AND-OR Circuits," in *Int. Workshop on Applications of the Reed-Muller Expansion in Circuit Design*, Aug. 1999.

[23] L. Wang and E. J. McCluskey, "Circuits for Pseudoexhaustive Test Pattern Generation," *IEEE Trans. on Computer Aided Design*, vol. 7, pp. 1068–1080, Oct. 1988.

[24] P. H. Bardell, W. H. McAnney, and J. Savir, *Built-in Test for VLSI: Pseudorandom Techniques*. New York: John Wiley and Sons, 1987.

[25] B. Konemann, J. Mucha, and G. Zwiehoff, "Built-In Logic Block Observation Technique," in *Digest of Papers 1979 Test Conference*, pp. 37–41, Oct. 1979.

[26] J. Saul, B. Eschermann, and J. Froessl, "Two-level Logic Circuits Using EXOR Sums of Products," *IEE Proceedings*, vol. 140, pp. 348–356, Nov. 1993.

[27] B. B. Bhattacharya, B. Gupta, S. Sarkar, and A. K. Choudhury, "Testable Design of RMC Networks with Universal Tests for Detecting Stuck-At and Bridging Faults," *IEE Proceedings*, vol. 132, pp. 155–161, May 1985.