Testing of stuck-open faults in generalised Reed–Muller and EXOR sum-of-products CMOS circuits

H. Rahaman, D.K. Das and B.B. Bhattacharya

Abstract: Testable designs of GRM (generalised Reed–Muller) and ESOP (EXOR sum-of-products) circuits have been proposed for robustly detecting all single stuck-open faults in their CMOS implementation. It is shown that for an n-variable boolean function, a sequence of $(4n + 13)$ vectors is sufficient to detect all single stuck-open faults in a GRM circuit. For an ESOP circuit, a test sequence of length $(2n + 10)$ is sufficient. In the first case, the EXOR part is designed as a tree of depth $\leq 2(\log\lceil p + 1 \rceil)$, and for the latter, as a linear cascade of length $\leq (p + 1)$, where p is the number of product terms in the GRM or ESOP expression. For both the cases the test sequence is universal, i.e. independent of the function and the circuit under test, and can be stored in a ROM for built-in self-test.

1 Introduction

In deep-submicron (DSM) designs, failure of a chip is often caused by several spot defects that cannot be modelled by the classical stuck-at fault model [1]. Interconnect 'breaks', which are prevalent in copper technology, and 'opens' are major sources of such defects [2]. During conventional chip testing it has been observed that most of the test escapes correspond to 'open' defects [3]. Additional tests are needed to detect their presence, which may be modelled by transition faults, delay faults, or stuck-open faults. For CMOS circuits, testing of stuck-open faults has been studied extensively in the past [4–13], and those techniques may be adapted to detect breaks/opens in the DSM regime. Since the test generation of stuck-open faults is difficult [7], various methods were reported for designing easily testable CMOS circuits [5,6], [8–13]. A CMOS combinational circuit implementing a Reed–Muller canonical (RMC) expression can be tested for stuck-open faults by a *universal* test [14]. A circuit realising an RMC expression generally suffers from slow speed and larger chip area. However, there are several variations of AND–EXOR based synthesis, which in many applications, may yield a lesser chip area than that of the traditional AND-OR synthesis [15]. They are also effectively used in the design of FPGA-based modules provided by Xilinx, Actel [16]. Further, these circuits are known to be easily testable. Various design-for-testability (DFT) techniques for detection of stuck-at faults by a universal test in AND–EXOR circuits are well known [17–20]. Testable realisation of RMC circuits for detecting bridging faults with a universal test set was described in [21]. For CMOS implementation of RMC expressions, a robust and universal test set for stuck-open faults was derived in [14]. Other notable works on optimal logic synthesis and other aspects of AND–EXOR based networks have appeared in [21–34].

Variations of AND–EXOR networks include positive-polarity Reed–Muller (PPRM) [17], fixed-polarity Reed–Muller (FPRM) [35], generalised Reed–Muller (GRM) [36], and EXOR–sum-of-products (ESOP) [18]. All the canonical Reed–Muller forms (PPRM, FPRM, and GRM) have certain restrictions on the polarities of variables or on product terms. Synthesis based on GRM and ESOP reduces the circuit cost compared with PPRM and FPRM [33]. Gate-level realisation of a GRM expression as well as that of an ESOP [19] is known to support a universal test set for stuck-at faults. Detection of bridging faults in GRM circuits has been studied recently [37].

This paper presents an easily testable CMOS implementation of a combinational circuit based on GRM and ESOP expressions for detecting all single stuck-open faults. A stuck-open fault in a combinational circuit may induce a sequential behaviour in the circuit. Testing of a stuck-open fault requires a two-pattern test, consisting of an initialisation and a test vector. Owing to the presence of arbitrary delays in the circuits, a two-pattern test may be *invalidated*. A two-pattern test, which cannot be invalidated, is called *robust*, and a circuit in which every irredundant stuck-open fault has a robust test pair, is called *robustly testable*. For detection of stuck-open faults in GRM and ESOP circuits, we slightly modify the design by using a few additional control inputs and one extra observable output. The modified circuit becomes robustly testable and admits a universal test sequence of length $(4n + 13)$ (resp. $(2n + 10)$) for a GRM (resp. ESOP) circuit, where n is the number of input variables. The test sequence can be stored in a ROM on chip for built-in self-testing (BIST). Thus, the control inputs may be treated as internal lines without having increased the number of external I/O pins.

2 Preliminaries

2.1 AND–EXOR logic expressions

Definition 1: An arbitrary boolean function $F(x_1, x_2, \ldots, x_n)$ can be represented as $F = a_0 \oplus a_1x_1 \oplus a_2x_2 \oplus \ldots \oplus a_nx_n \oplus a_{12}x_1x_2 \oplus a_{13}x_1x_3 \ldots \oplus a_{(n-1)n}x_{n-1}x_n \oplus \ldots \oplus a_{12\ldots n}x_1x_2x_3 \ldots x_n$. Such an expression is called positive-polarity Reed-Muller (PPRM), in which only positive polarities are allowed for each input variable. The coefficients $a_0, a_1, a_2, \ldots, a_{12\ldots n}$ are unique for a given function. For example, $x_1x_2 \oplus x_2x_3$ is a PPRM expression since all the variables appear with only positive polarities.

Definition 2: If we use either a positive literal x_i or negative literal $\bar{x}_i$ in the preceding expression, it is called a fixed polarity Reed–Muller expression (FPRM). For example, $\bar{x}_1x_2 \oplus x_2\bar{x}_3$ belongs to this class.

Definition 3: If an AND–EXOR expression has no restrictions on the polarities of variables, but does not allow the same set of variables to appear in more than one product term, it is called generalised Reed–Muller expression (GRM). It is also known as canonical restricted polarity form (CRMP). For example, $\bar{x}_1 \oplus \bar{x}_2 \oplus x_2\bar{x}_3$ is a GRM.

Definition 4: An expression consisting of arbitrary product terms combined by EXOR operations is called an EXOR sum-of-products expression (ESOP). For example, $x_1x_2\bar{x}_3 \oplus \bar{x}_1x_2x_3$ is an ESOP, but not a GRM, as the same set of variables $\{x_1, x_2, x_3\}$ appears in two product terms.

It has been observed that ESOPs usually require fewer product terms than GRMs; PPRMs often need more product terms than GRMs. For many boolean functions, GRMs require fewer product terms than the classical AND–OR SOPs [33]. Table 1 shows data on some benchmark circuits.

Table 1: Comparison of number of products terms

		Number of products terms		
Circuit	Inputs	GRM	ESOP	SOP
adr5	10	67	63	167
adr9	18	1031	1023	3031
adr10	20	2056	2047	6099
b9	16	81	81	119
in2	19	134	122	108
inc13	13	25	25	92
intb	15	375	327	629
m181	15	29	29	41
misex2	25	27	27	28
mlp6	12	1277	862	1870
rd73	7	63	35	127
rd84	8	107	58	255
sao2	10	28	28	58
sqr6	6	35	34	47
sym10	10	110	82	210
sym9	9	126	51	84
t1	21	94	90	103
t481	16	13	13	481
wgt12	12	1068	445	4095

2.2 Testability properties of stuck-at faults in AND–EXOR networks

PPRM realisation: For an n-variable function, Reddy proposed an EXOR cascade implementation of PPRM expression [17]. This structure admits a universal test set of length $(n + 4)$ for detecting all single stuck-at faults in the circuit.

GRM realisation: Sasao proposed a testable design for easy detection of multiple stuck-at faults in GRM gate-level circuits [36]. The proposed design consists of four parts: literal part, AND part, EXOR part, and check part. The EXOR part is implemented as a tree structure instead of a cascade to reduce circuit delay. However, the design does not admit a universal test set.

ESOP realisation: Pradhan proposed an ESOP circuit structure for easy detection of multiple stuck-at faults [18]. Another testable implementation was reported later [19] for which a universal test set of length $(n + 6)$ exists that detects all single stuck-at faults.

2.3 Stuck-open faults in CMOS circuits

A permanent disconnection between source and drain of a transistor is modelled by a stuck-open fault [38]. Under this fault, i.e. in the presence of a nonconducting transistor, the output of the circuit under certain input may float and show the previous logic value. Thus in faulty condition, a combinational circuit may behave as a sequential machine. Detection of this fault requires a two-pattern test consisting of an initialisation vector $\boldsymbol{X_I}$ and a test vector $\boldsymbol{X_T}$.

Definition 5: A test $\boldsymbol{X_T}$ for a stuck-open fault in a CMOS combinational circuit realising a nonconstant function $\boldsymbol{F}$ is a vector, on application of which the output becomes floating in the presence of the fault.

Definition 6: An initialisation vector $\boldsymbol{X_I}$ for a stuck-open fault is an input, such that $F(\boldsymbol{X_I})$ is the complement of $F(\boldsymbol{X_T})$, in the fault-free condition. Owing to the presence of unequal delays in the circuit, some transient vector may appear at the inputs of the circuit when $\boldsymbol{X_T}$ is applied following $\boldsymbol{X_I}$. This may invalidate initialisation and thus fail to detect the fault.

Definition 7: A two-pattern test that cannot be invalidated is called robust, else it is nonrobust.

It is known that for some stuck-open faults in a circuit, robust tests may not exist [5]. If the Hamming distance between $\boldsymbol{X_I}$ and $\boldsymbol{X_T}$ is greater than one, a robust test may not exist.

2.3.1 PPRM realisation to detect stuck-open faults:

To derive a universal test sequence for stuck-open faults in CMOS implementation, the PPRM network was modified by using NAND gates instead of AND gates [14]. If all the literals appear an odd number of times in the PPRM expression, the design requires only one control input. If some literals appear an even number of times, an additional control input and two extra gates are needed. It admits a universal test sequence of length $(2n + 8)$ for robust detection of all single stuck-open faults in an n-input CMOS circuit.

2.3.2 Testing of NAND, NOR, and EXOR CMOS gates:

The following lemmata are self-evident. A CMOS realisation of a NAND (NOR) gate is shown in Fig. 1*a* (Fig. 1*b*). We assume that faulty transistor (stuck-open) is fed by the literal x_j.

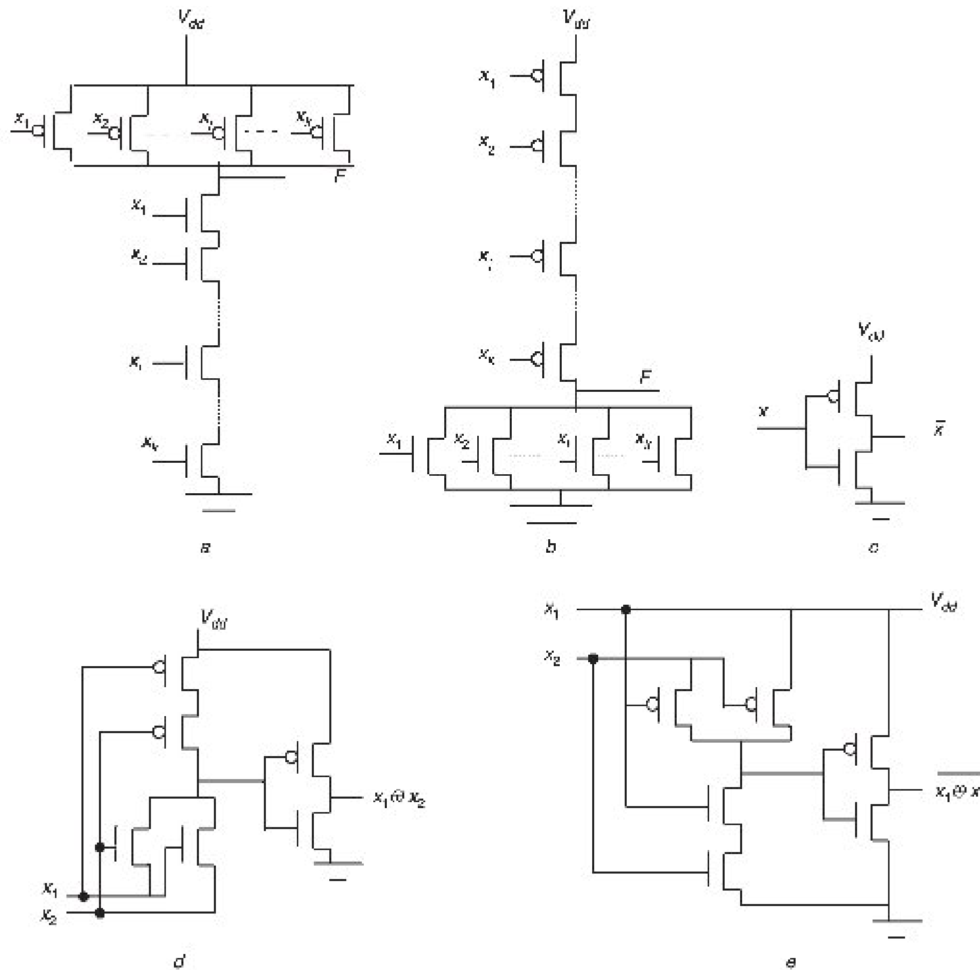


Fig. 1 *Testing stuck-open faults in CMOS gates*

a NAND
b NOR
c NOT
d EXOR
e EXNOR

Lemma 1: The stuck-open fault in a *p*-transistor [resp. *n*-transistor] of a NAND gate can be robustly detected by the following two-pattern test:

$$\{x_1, x_2, \ldots, x_j, \ldots, x_k\} = \{(1, 1, \ldots, 1, \ldots, 1), (1, 1, \ldots, 0, \ldots, 1)\}$$
$$[\text{resp.} \{(1, 1, \ldots, 0, \ldots, 1), (1, 1, \ldots, 1, \ldots, 1)]$$

Lemma 2: The stuck-open fault in a *p*-transistor of a NOR gate can be robustly detected by the following two-pattern test:

$$\{x_1, x_2, \ldots, x_j, \ldots, x_k\} = \{(0, 0, \ldots, 1, \ldots, 0), (0, 0, \ldots, 0, \ldots, 0)\}$$
$$[\text{resp.} \{(0, 0, \ldots, 0, \ldots, 0), (0, 0, \ldots, 1, \ldots, 0)\}]$$

If a NAND or NOR gate has only one input, it behaves as a NOT gate as shown in Fig. 1c.

Lemma 3: Any single stuck-open fault in a NOT gate is robustly tested by the test sequence {0, 1, 0}.

For synthesis of EXOR and EXNOR gates we use the special structure as in [4], shown in Figs. 1*d* and 1*e*.

Lemma 4: [4] Any single stuck-open fault in the EXOR gate of Fig. 1*d* can be robustly detected by the following three two-pattern tests: {00 or 11, 01}, {00 or 11,10}, {01 or 10, 00}.

Lemma 5: [4] Any single stuck-open in the EXNOR gate of Fig. 1*e* can be robustly detected by the following three two-pattern tests: {00 or 11, 01}, {00 or 11,10}, {01 or 10, 11}.

3 Proposed design technique

3.1 GRM realisations

We propose a testable design for robust detection of all single stuck-open faults by a universal test in a CMOS circuit based on GRM realisation.

3.1.1 Basic scheme:

The design consists of four parts: literal part, NAND part, EXOR part, and check part. The literal part consists of EXOR gates with a control line and is used to produce the positive (uncomplemented) or negative (complemented) literals. The proposed scheme is similar to Sasao's design [36], but with three differences: the AND part is replaced by a NAND array with one control line, the check part has only one observable output instead of four, and the EXOR tree is redesigned with two control inputs. Four control lines are thus needed; C_1 in the literal part, C_2 in the NAND part, C_3 and C_4 in the EXOR part.

Our scheme is based on the following steps.

Step 1: Each AND term in the GRM expression is replaced by NAND for ease of synthesis.

Example 1: Consider a GRM expression for a function $F_1 = \bar{x}_1 \oplus \bar{x}_1\bar{x}_2 \oplus x_2\bar{x}_3$. In terms of NAND operations, we write $F_1 = \bar{x}_1 \oplus \overline{\bar{x}_1\bar{x}_2} \oplus \overline{x_2\bar{x}_3}$. If a GRM expression consists of odd number of AND terms, the whole expression is complemented by EXOR-ing it with constant function 1.

Example 2: Consider a function $F_2 = \bar{x}_1x_2 \oplus x_2\bar{x}_3 \oplus x_1x_3$. In terms of NAND operations, we write $F_2 = 1 \oplus \overline{\bar{x}_1x_2} \oplus \overline{x_2\bar{x}_3} \oplus \overline{x_1x_3}$.

Example 3: Consider a GRM expression for a function $F_3 = x_1 \oplus x_1\bar{x}_2 \oplus x_2\bar{x}_3$. In this case, we complement x_1 and write, $F_3 = 1 \oplus \bar{x}_1 \oplus \overline{x_1\bar{x}_2} \oplus \overline{x_2\bar{x}_3}$.

Step 2: The literals that appear even number of times in the GRM expression are used to feed a new NAND gate with a control input C_2. To realise the original function by setting $C_2 = 0$, we complement the expression as obtained in step-1.

Example 4: In the function F_1 of example 1, the literals x_1 and x_2 appear even number of times. Thus F_1 is changed to $F'_1 = 1 \oplus \bar{x}_1 \oplus \overline{\bar{x}_1\bar{x}_2} \oplus \overline{x_2\bar{x}_3} \oplus \overline{C_2x_1x_2}$.

Example 5: In the function F_3 of example 3, the literals x_1 and x_2 appear even number of times. F_3 is changed to $F'_3 = 1 \oplus 1 \oplus \bar{x}_1 \oplus \overline{x_1\bar{x}_2} \oplus \overline{x_2\bar{x}_3} \oplus \overline{C_2x_1x_2} = \bar{x}_1 \oplus \overline{\bar{x}_1\bar{x}_2} \oplus \overline{x_2\bar{x}_3} \oplus \overline{C_2x_1x_2}$. By putting $C_2 = 0$ in F'_3, and we get $F'_3 = F_3$.

Step 3: If the constant function 1 appears in the expression, we eliminate it by changing an EXOR operation with an EXNOR operation.

Example 6: The function F'_1, shown is expressed as $F'_1 = \bar{x}_1 \bar{\oplus} \overline{\bar{x}_1\bar{x}_2} \oplus \overline{x_2\bar{x}_3} \oplus \overline{C_2x_1x_2}$, where $\bar{\oplus}$ represents an EXNOR operation.

Step 4: Following the expression obtained in step 3, we synthesise the circuit as shown in Fig. 2. Two control inputs C_3 and C_4 are used in the EXOR block. For inverting a single literal, we use a NOT gate. To realise the basic NAND, NOR, NOT gates in CMOS, we use the structures of Figs. 1*a*, 1*b*, and 1*c*, respectively. The EXOR operations in step-3 are implemented in the form of tree. The EXNOR operation if any, appears as the last gate in the EXOR block.
Step 5: Each EXOR gate (or EXNOR, if any) is replaced by two EXOR gates and a control input (either C_3 or C_4) as shown in Fig. 3. These two control inputs are used in alternate levels. All EXOR (EXNOR) gates are implemented as shown in Fig. 1*d* (1*e*). Since in step 2

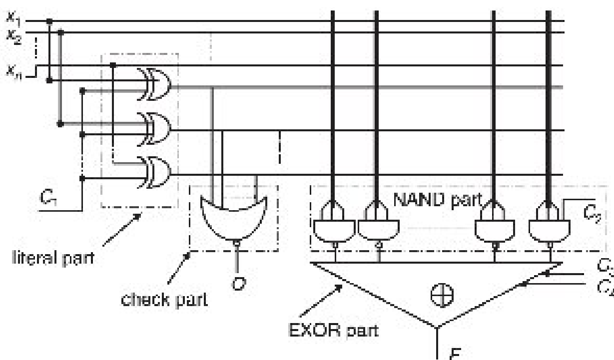


Fig. 2 *Stuck-open fault testable GRM network*

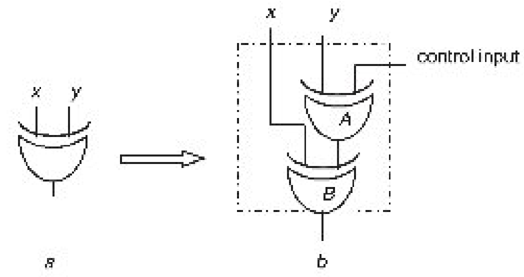


Fig. 3 *EXOR gates*
a Original
b Redesigned

an extra NAND gate may be required, the depth of the tree will be $\leq 2(\log\lceil p+1 \rceil)$, where p is the number of products terms in the GRM. The design requires four control inputs and one extra observable output.

Example 7: An EXOR tree structure (Fig. 4*a*) is obtained after step 3. This is redesigned as in Fig. 4*b* with two control inputs C_3 and C_4. By setting these control inputs to 0, the original function can be obtained.

Example 8: The circuit for the function of example 6 is shown in Fig. 5.

3.1.2 Robust test set:

If we put $C_3 = C_4 = 0$, the output function is same as the original function. When $C_1 = 0$, the expression becomes a PPRM as shown in [14]. The following theorem is easy to prove.

Theorem 1: Every single stuck-open fault in the transistors realising the NAND gates of a NAND-EXOR realisation (Fig. 2) has a robust test.

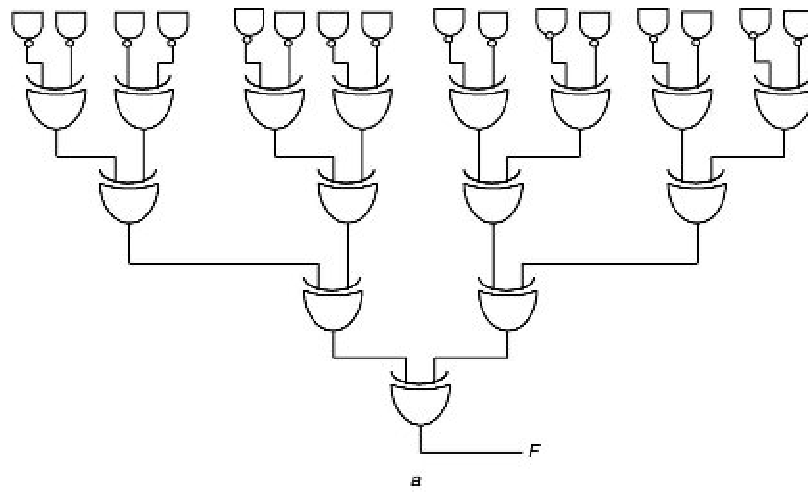
Lemma 6: Any single stuck-open fault in the NAND gates (Fig. 2) is robustly testable by the following universal test sequence of length $2n+3$, where n is the number of variables:

$$T_1 = \left|\begin{array}{cccccc}
C_1 & C_2 & C_3 & C_4 & x_1 & x_2 \ldots\ldots x_j \ldots x_n \\
0 & 1 & 0 & 0 & 1 & 1 \ldots\ldots 1 \ldots\ldots 1 \\
0 & 0 & 0 & 0 & 1 & 1 \ldots\ldots 1 \ldots\ldots 1 \\
0 & 1 & 0 & 0 & 1 & 1 \ldots\ldots 1 \ldots\ldots 1 \\
 & & & & & \\
0 & 1 & 0 & 0 & 0 & 1 \ldots\ldots 1 \ldots\ldots 1 \\
0 & 1 & 0 & 0 & 1 & 1 \ldots\ldots 1 \ldots\ldots 1 \\
0 & 1 & 0 & 0 & 1 & 0 \ldots\ldots 1 \ldots\ldots 1 \\
0 & 1 & 0 & 0 & 1 & 1 \ldots\ldots 1 \ldots\ldots 1 \\
\vdots & & & & & \\
\vdots & & & & & \\
0 & 1 & 0 & 0 & 1 & 1 \ldots\ldots 0 \ldots\ldots 1 \\
0 & 1 & 0 & 0 & 1 & 1 \ldots\ldots 1 \ldots\ldots 1 \\
0 & 1 & 0 & 0 & 1 & 1 \ldots\ldots 1 \ldots\ldots 0 \\
0 & 1 & 0 & 0 & 1 & 1 \ldots\ldots 1 \ldots\ldots 1
\end{array}\right|$$

Lemma 7: The test for stuck-open fault in the NOT gate, if any, is included in T_1.

Proof: Follows directly from lemma 3. □

The check part of the proposed GRM circuit includes a NOR gate. For detection of stuck-open faults in the NOR gate, responses to the test vectors will be observed at the additional output line O.



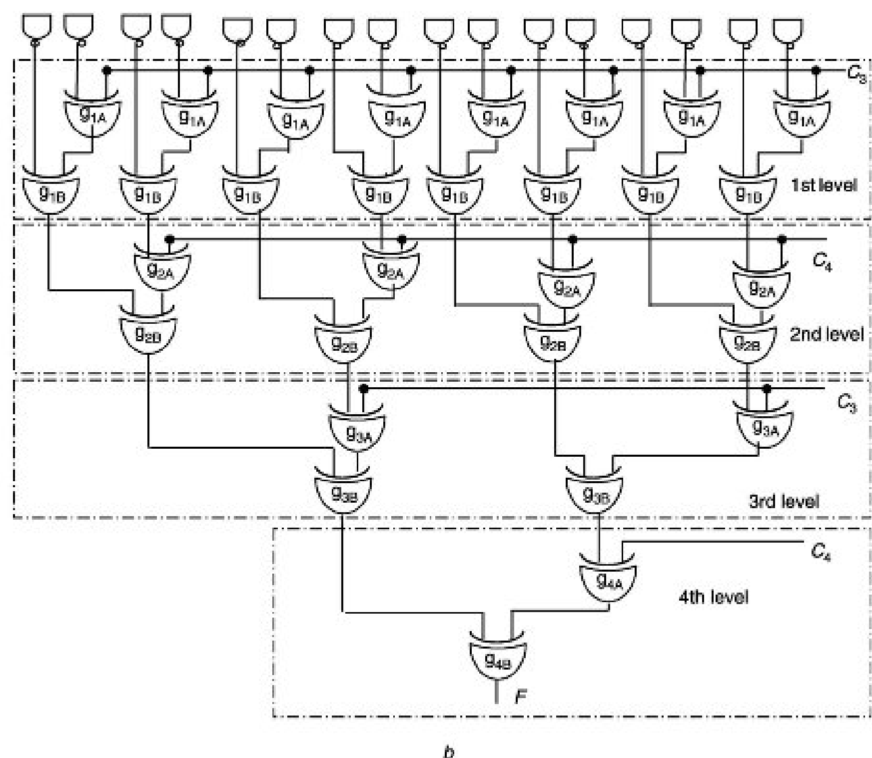


Fig. 4 *EXOR tree structure*
a Example
b Stuck-open testable design

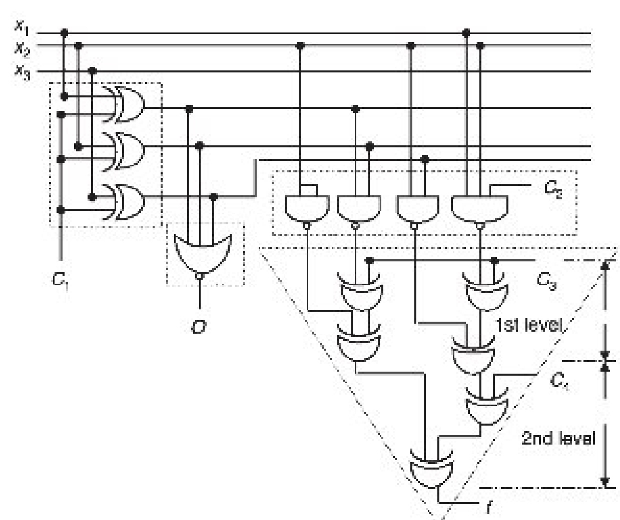


Fig. 5 *Example of stuck-open testable GRM network*

Lemma 8: Any single stuck-open fault in the NOR gate is robustly testable by the following universal test sequence T_2 of length $2n+1$:

$$T_2 = \left| \begin{array}{cccccl} C_1 & C_2 & C_3 & C_4 & x_1 & x_2 \ldots\ldots x_j \ldots x_n \\ 1 & - & 0 & 0 & 1 & 1\ldots\ldots 1 \ldots\ldots 1 \\ 1 & - & 0 & 0 & 0 & 1\ldots\ldots 1 \ldots\ldots 1 \\ 1 & - & 0 & 0 & 1 & 1\ldots\ldots 1 \ldots\ldots 1 \\ 1 & - & 0 & 0 & 1 & 0\ldots\ldots 1 \ldots\ldots 1 \\ 1 & - & 0 & 0 & 1 & 1\ldots\ldots 1 \ldots\ldots 1 \\ \vdots & & & & & \\ \vdots & & & & & \\ 1 & - & 0 & 0 & 1 & 1\ldots\ldots 0 \ldots\ldots 1 \\ 1 & - & 0 & 0 & 1 & 1\ldots\ldots 1 \ldots\ldots 1 \\ 1 & - & 0 & 0 & 1 & 1\ldots\ldots 1 \ldots\ldots 0 \\ 1 & - & 0 & 0 & 1 & 1\ldots\ldots 1 \ldots\ldots 1 \end{array} \right|$$

Proof: Since $C_1 = 1$, the inputs to the NOR gates are $(\bar{x}_1, \bar{x}_{2,\ldots\ldots}\bar{x}_i, \ldots\ldots \bar{x}_n)$. The following two consecutive test

patterns in $T_2(C_1, C_2, C_3, C_4, x_1, x_2, \ldots, x_i \ldots, x_n)$: $[\{1, d, 0, 0, 1, 1, \ldots, 0, \ldots 1\}, \{1, d, 0, 0, 1, 1, \ldots, 1, \ldots, 1\}]$ produce $(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_i, \ldots, \bar{x}_n) = [\{0, 0, \ldots, 1, \ldots, 0\}, \{0, 0, \ldots, 0, \ldots, 0\}]$ at the inputs of the NOR gate. Hence, they robustly detect the stuck-open fault in the p-transistor fed by the literal x_i by lemma 2. Similarly, the two consecutive patterns in $T_2(C_1, C_2, C_3, C_4, x_1, x_2, \ldots, x_i \ldots, x_n)$: $[\{1, d, 0, 0, 1, 1, \ldots, 1, \ldots, 1\}, \{1, d, 0, 0, 1, 1, \ldots, 0, \ldots, 1\}]$ robustly detect the stuck-open fault in the n-transistor fed by x_i. It can also be proved that the test sequence for any other single stuck-open fault in the NOR gate is included in T_2. □

Lemma 9: Any single stuck-open fault in the EXOR gate (fed by x_j) in the literal part can be robustly detected by the following five-pattern test sequence T_3. For the first three patterns the response is observed at the functional output F and for the last two patterns the response is observed at the output O.

$$T_3 = \begin{vmatrix} C_1 & C_2 & C_3 & C_4 & x_1 & x_2 \ldots\ldots x_j \ldots\ldots x_n \\ 0 & 1 & 0 & 0 & 1 & 1 \ldots\ldots 1 \ldots\ldots 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \ldots\ldots 0 \ldots\ldots 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \ldots\ldots 1 \ldots\ldots 1 \\ \\ 1 & 1 & 0 & 0 & 1 & 1 \ldots\ldots 1 \ldots\ldots 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \ldots\ldots 0 \ldots\ldots 1 \end{vmatrix}$$

Proof: Application of the following two consecutive test patterns $(C_1, C_2, C_3, C_4, x_1, x_2, , x_j, \ldots, x_n) = \{(0, 1, 0, 0, 1, 1, \ldots, 1, \ldots, 1), (0, 1, 0, 0, 1, 1, \ldots, 0, 1)\}$, produces the sequence (10, 00) at the inputs of the EXOR gate fed by x_j and similarly the two-pattern test $\{(0, 1, 0, 0, 1, 1, \ldots, 0, \ldots, 1), (0, 1, 0, 0, 1, 1, \ldots, 1, \ldots, 1)\}$, produces (00, 10). Further, $\{(1, 1, 0, 0, 1, 1, \ldots, 1, \ldots, 1), (1, 1, 0, 0, 1, 1, \ldots 0, \ldots, 1)\}$, produces (11, 01). Thus, the EXOR gate connected to x_j in the literal part receives three two-pattern vector pairs: [10, 00], [00, 10], [11, 01] that are needed for robustly testing all stuck-open faults in it. For the first three patterns, the effects of the fault are propagated to F. For the last pair, the effect of the can be propagated to the observable point O. Hence, by lemma 4, the EXOR gate is tested. □

Lemma 10: Any single stuck-open fault in the literal part of the network is robustly tested by the test sequence T_L, where $T_L = T_1 \cup T_2$.

Proof: Follows from the fact that T_3 is included in $T_1 \cup T_2$.□

Lemma 11: Any single stuck-open fault in the EXOR block of the network is robustly testable by the following universal test sequence T_4 consisting of 10 patterns:

$$T_4 = \begin{vmatrix} C_1 & C_2 & C_3 & C_4 & x_1 & x_2 \ldots\ldots x_j \ldots\ldots x_n \\ 0 & 1 & 0 & 0 & 1 & 1 \ldots\ldots 1 \ldots\ldots 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \ldots\ldots 1 \ldots\ldots 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \ldots\ldots 1 \ldots\ldots 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \ldots\ldots 1 \ldots\ldots 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \ldots\ldots 1 \ldots\ldots 1 \\ \\ 0 & - & 1 & 1 & 0 & 0 \ldots\ldots 0 \ldots\ldots 0 \\ 0 & - & 1 & 0 & 0 & 0 \ldots\ldots 0 \ldots\ldots 0 \\ 0 & - & 1 & 1 & 0 & 0 \ldots\ldots 0 \ldots\ldots 0 \\ 0 & - & 0 & 1 & 0 & 0 \ldots\ldots 0 \ldots\ldots 0 \\ 0 & - & 1 & 1 & 0 & 0 \ldots\ldots 0 \ldots\ldots 0 \end{vmatrix}$$

Proof: For the first five patterns in T_4, the outputs of NAND gates (or NOT gates, if any) are 0. When the first vector is applied, all gates in the tree receive an input 00 in the fault-free condition. Similarly, for the third and fifth pattern, each gate in the tree receives 00 (see Fig. 4*b*).

Application of the second vector in T_4 produces 00 at the inputs of all A-gates and B-gates lying in the first level. Thus, the output of each gate in this level becomes 0. In the second level, the control input C_4 is set to 1, and hence all A- and B-gates in the second level receive 01 producing a 1 output from each such gate. In the third level, the control input C_3 is set to 0, which produces 10 to the inputs of A-gates, 11 to B-gates, and consequently, the output of each gate in this level becomes 0. Gates at other even-numbered levels (i.e. fourth, sixth, …) behave similarly. The behaviour of the gates at odd-numbered levels (i.e. fifth, seventh, …) is same as that of the third level. Similarly, we can show that the fourth pattern in T_4 produces (i) 01 to inputs of each A and B-gate lying in an odd-numbered level, (ii) 10 to each A-gate and 11 to each B-gate lying in an even-numbered level.

For the last five patterns in T_4, the output of each NAND gate (or NOT gate, if any) is 1. Application of sixth, eighth and tenth patterns in T_4, produces 11 to each A-gate and 10 to each B-gate in all levels. Application of the seventh pattern in T_4 generates (i) 11 to each A-gate and 10 to each B-gate in the first level, (ii) 10 to each A-gate and 11 to each B gate of second, fourth, sixth, …, levels, (iii) 01 to each A-gate and B-gate in third, fifth, seventh ,…, levels. Similarly application of ninth pattern produces (i) 10 to each A-gate and 11 to each B-gate at odd-numbered levels, (ii) 01 to each A-gate and each B-gate of even-numbered levels.

In summary, the first five patterns in T_4, produce the following sequence: (i) (00, 00, 00, 01, 00) to each A-gate and B-gate in the first level, (ii) (00, 01, 00, 10, 00) to each A-gate and (00, 01. 00, 11, 00) to each B-gate in the second, fourth, sixth …, level, (iii) (00, 10, 00, 01, 00) to each A-gate and (00, 11, 00, 01, 00) to each B-gate in the third, fifth, seventh …, level. Similarly, the last five patterns in T_4, generate (i) (11, 11, 11, 10, 11) to each A-gate and (10, 10, 10, 11, 10) to each B-gate in the first level, (ii) (11, 10, 11, 01, 11) to each A-gate and (10, 11, 10, 01, 10) to each B-gate in the second, fourth, sixth, …, level, (iii) (10, 11, 10, 01, 10) to each A-gate and (10, 01, 10, 11, 10) to each B-gate in the third, fifth, seventh …, level. Therefore each gate in the tree receives the inputs (00, 01), (01, 00), (11, 10), (10, 11). By lemma 3, it follows that each EXOR/EXNOR gate is robustly tested by T_4. □

Theorem 2: Any stuck-open fault in the GRM network is robustly testable by the universal test sequence T of length $(4n + 13)$, where n is the number of variables, $T = T_1 \cup T_2 \cup T_4$.

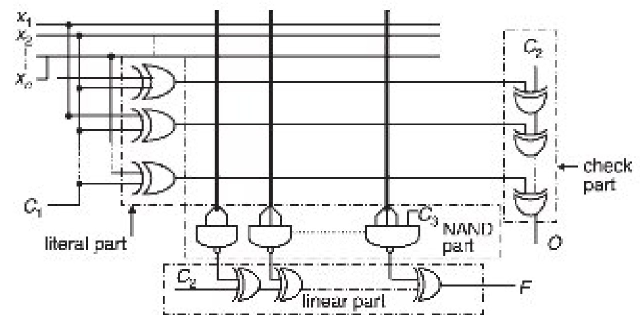


Fig. 6 *Stuck-open testable ESOP circuit*

$$
T = \left|\begin{array}{cccccl}
C_1 & C_2 & C_3 & C_4 & x_1 & x_2\ldots\ldots x_j\ldots x_n \\
0 & 1 & 0 & 0 & 1 & 1\ldots\ldots1\ldots\ldots1 \\
0 & 1 & 0 & 1 & 1 & 1\ldots\ldots1\ldots\ldots1 \\
0 & 1 & 0 & 0 & 1 & 1\ldots\ldots1\ldots\ldots1 \quad \} 5 \\
0 & 1 & 1 & 0 & 1 & 1\ldots\ldots1\ldots\ldots1 \\
0 & 1 & 0 & 0 & 1 & 1\ldots\ldots1\ldots\ldots1 \\
 & & & & & \\
0 & 0 & 0 & 0 & 1 & 1\ldots\ldots1\ldots\ldots1 \\
0 & 1 & 0 & 0 & 1 & 1\ldots\ldots1\ldots\ldots1 \\
0 & 1 & 0 & 0 & 0 & 1\ldots\ldots1\ldots\ldots1 \\
\vdots & & & & & \qquad\qquad\quad\vdots \\
0 & 1 & 0 & 0 & 1 & 1\ldots\ldots1\ldots\ldots1 \\
0 & 1 & 0 & 0 & 1 & 1\ldots\ldots0\ldots\ldots1 \quad \} 2n+2 \\
\vdots & & & & & \qquad\qquad\quad\vdots \\
0 & 1 & 0 & 0 & 1 & 1\ldots\ldots1\ldots\ldots1 \\
0 & 1 & 0 & 0 & 1 & 1\ldots\ldots1\ldots\ldots0 \\
0 & 1 & 0 & 0 & 1 & 1\ldots\ldots1\ldots\ldots1 \\
 & & & & & \\
1 & 1 & 0 & 0 & 1 & 1\ldots\ldots1\ldots\ldots1 \\
1 & 1 & 0 & 0 & 0 & 1\ldots\ldots1\ldots\ldots1 \\
1 & 1 & 0 & 0 & 1 & 1\ldots\ldots1\ldots\ldots1 \\
1 & 1 & 0 & 0 & 1 & 0\ldots\ldots1\ldots\ldots1 \\
\vdots & & & & & \qquad\qquad\quad\vdots \\
1 & 1 & 0 & 0 & 1 & 1\ldots\ldots1\ldots\ldots1 \quad \} 2n+1 \\
1 & 1 & 0 & 0 & 1 & 1\ldots\ldots0\ldots\ldots1 \\
\vdots & & & & & \qquad\qquad\quad\vdots \\
1 & 1 & 0 & 0 & 1 & 1\ldots\ldots1\ldots\ldots1 \\
1 & 1 & 0 & 0 & 1 & 1\ldots\ldots1\ldots\ldots0 \\
1 & 1 & 0 & 0 & 1 & 1\ldots\ldots1\ldots\ldots1 \\
 & & & & & \\
0 & - & 1 & 1 & 0 & 0\ldots\ldots0\ldots\ldots0 \\
0 & - & 1 & 0 & 0 & 0\ldots\ldots0\ldots\ldots0 \\
0 & - & 1 & 1 & 0 & 0\ldots\ldots0\ldots\ldots0 \quad \} 5 \\
0 & - & 0 & 1 & 0 & 0\ldots\ldots0\ldots\ldots0 \\
0 & - & 1 & 1 & 0 & 0\ldots\ldots0\ldots\ldots0
\end{array}\right|
$$

Proof: Follows from lemmas 6, 8, 10 and 11. □

In T, the first five patterns are taken from the first half of T_4, the last vector of which is same as the first vector of T_1. The sequence T_2 and the last part of T_4 are concatenated next.

Since the proposed design admits a universal test sequence of two-pattern tests, it simplifies the test generation process and thus no ATPG is required. In most of the cases, it compares favourably with traditional two-level AND-OR synthesis. For example the circuit *wgt12* (Table 1) consists of 4095 product terms in the s-o-p expression and has 12 inputs. The proposed design will need a test sequence of 61 vectors for complete and robust testing of all single stuck-open faults. Sasao's GRM realisation [7] will require $\{m + 2n + 3\} = 1095$ test vectors for detecting all single stuck-at faults, where m is the number of GRM product terms (= 1068 for the circuit *wgt12*). The test sequence T can be stored in a ROM of size = $(4n + 13) \times (n + 4)$ bits.

3.2 Stuck-open testable ESOP network

In the proposed scheme each AND operation in an ESOP expression is replaced by a NAND operation for ease of synthesis. The design is similar to the scheme proposed earlier for detection of stuck-at faults [19]. The structure consists of literal part, NAND part, check part, and EXOR (linear) part as shown in Fig. 6. The check part consists of an EXOR cascade with an extra observable output O. The check part is used to test the literal part.

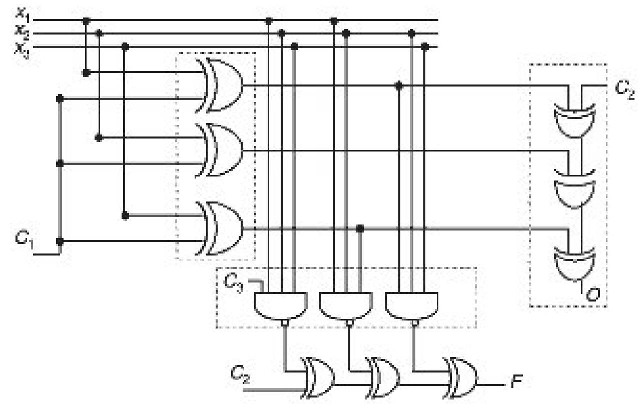


Fig. 7 *Example of stuck-open testable ESOP circuit*

Three control inputs are needed: C_1 in the literal part, C_2 in the check and linear part, and C_3 in the NAND part. The linear part is implemented with an EXOR cascade. Each EXOR gate is realised with a special type of CMOS structure as shown in Fig. 1*d*. If some literals in the ESOP expression appear an even number of times, the control input C_3, an extra NAND gate and an EXOR are required. The literals, which appear an even number of times, are fed to this NAND gate along with the control C_3. An EXOR gate is needed to combine this term in the cascade. An example of stuck-open testable ESOP realisation for the function F_1 is shown in Fig. 7.

3.3 Universal and robust test set for ESOP realisation

For $C_1 = 0$, the ESOP circuit is same as a PPRM circuit [14].

Lemma 12: If each EXOR gate in the linear part is realised with a CMOS circuit of Fig. 1*d*, the linear part of the ESOP network is robustly testable by the following universal test sequence (T_5) of constant length 6:

$$
T_5 = \left|\begin{array}{ccccl}
C_1 & C_2 & C_3 & x_1 & x_2\ldots\ldots x_j\ldots\ldots x_n \\
0 & 0 & 0 & 0 & 0\ldots\ldots0\ldots\ldots0 \\
0 & 1 & 0 & 0 & 0\ldots\ldots0\ldots\ldots0 \\
0 & 0 & 0 & 0 & 0\ldots\ldots0\ldots\ldots0 \\
0 & 1 & 1 & 1 & 1\ldots\ldots1\ldots\ldots1 \\
0 & 0 & 1 & 1 & 1\ldots\ldots1\ldots\ldots1 \\
0 & 1 & 1 & 1 & 1\ldots\ldots1\ldots\ldots1
\end{array}\right|
$$

It is shown in [14] that the NAND part of the network is robustly tested by a universal test sequence of length $2n + 3$. The output response is observed at the functional output F.

Lemma 13: Any single stuck-open fault in NAND gates of an ESOP network is robustly testable by the following universal test sequence (T_6) of length $2n + 3$:

$$
T_6 = \left|\begin{array}{ccccl}
C_1 & C_2 & C_3 & x_1 & x_2\ldots\ldots x_j\ldots\ldots x_n \\
0 & \mathrm{d} & 1 & 1 & 1\ldots\ldots1\ldots\ldots1 \\
0 & \mathrm{d} & 0 & 1 & 1\ldots\ldots1\ldots\ldots1 \\
0 & \mathrm{d} & 1 & 1 & 1\ldots\ldots1\ldots\ldots1 \\
0 & \mathrm{d} & 1 & 0 & 1\ldots\ldots1\ldots\ldots1 \\
0 & \mathrm{d} & 1 & 1 & 1\ldots\ldots1\ldots\ldots1 \\
\vdots & & & & \qquad\qquad\quad\vdots \\
0 & \mathrm{d} & 1 & 1 & 1\ldots\ldots0\ldots\ldots1 \\
0 & \mathrm{d} & 1 & 1 & 1\ldots\ldots1\ldots\ldots1 \\
0 & \mathrm{d} & 1 & 1 & 1\ldots\ldots1\ldots\ldots0 \\
0 & \mathrm{d} & 1 & 1 & 1\ldots\ldots1\ldots\ldots1
\end{array}\right|
$$

Lemma 14: The literal part of an ESOP network is robustly testable by the two-pattern test sequence $T_L = T_6 \cup T_7$, where T_7 is given by

$$T_7 = \left| \begin{array}{ccccc} C_1 & C_2 & C_3 & x_1 & x_2 \ldots\ldots x_j \ldots\ldots x_n \\ 0 & \mathrm{d} & 0 & 0 & 0 \ldots\ldots 0 \ldots\ldots 0 \\ 1 & \mathrm{d} & 0 & 0 & 0 \ldots\ldots 0 \ldots\ldots 0 \end{array} \right|$$

Proof: Application of test patterns in T_6 produces [10, 00] and [00, 10] to the inputs of all EXOR gates in the literal part. Similarly, when T_7 is applied, all EXOR gates receive [00, 01]. By observing the response at the output O, the fault can be detected.

Lemma 15: The check part is robustly testable by T_5, a test sequence of constant length 6.

Proof: The first three vectors of T_5 produce (00, 01, 00) to the inputs of each EXOR gate. Similarly, the last three vectors of T_5 produce (11, 10, 11). By observing the response at O, we detect any single stuck-open fault in this part. □

Theorem 3: Any stuck-open fault in the ESOP network designed as in Fig. 6 is robustly testable by the following universal test sequence T of length $2n + 10$:

$$T = \left| \begin{array}{ccccc} C_1 & C_2 & C_3 & x_1 & x_2 \ldots\ldots x_j \ldots\ldots x_n \\ 0 & 1 & 1 & 1 & 1 \ldots\ldots 1 \ldots\ldots 1 \\ 0 & 0 & 1 & 1 & 1 \ldots\ldots 1 \ldots\ldots 1 \\ 0 & 1 & 1 & 1 & 1 \ldots\ldots 1 \ldots\ldots 1 \\ 0 & 1 & 0 & 1 & 1 \ldots\ldots 1 \ldots\ldots 1 \\ 0 & 1 & 1 & 1 & 1 \ldots\ldots 1 \ldots\ldots 1 \\ 0 & 1 & 1 & 0 & 1 \ldots\ldots 1 \ldots\ldots 1 \\ 0 & 1 & 1 & 1 & 1 \ldots\ldots 1 \ldots\ldots 1 \\ 0 & 1 & 1 & 1 & 0 \ldots\ldots 1 \ldots\ldots 1 \\ \ldots & \ldots & \ldots & & \\ 0 & 1 & 1 & 1 & 1 \ldots\ldots 0 \ldots\ldots 1 \\ 0 & 1 & 1 & 1 & 1 \ldots\ldots 1 \ldots\ldots 1 \\ \ldots & \ldots & \ldots & & \\ 0 & 1 & 1 & 1 & 1 \ldots\ldots 1 \ldots\ldots 1 \\ 0 & 1 & 1 & 1 & 1 \ldots\ldots 1 \ldots\ldots 0 \\ 0 & 1 & 1 & 1 & 1 \ldots\ldots 1 \ldots\ldots 1 \\ 0 & 0 & 0 & 0 & 0 \ldots\ldots 0 \ldots\ldots 0 \\ 1 & 0 & 0 & 0 & 0 \ldots\ldots 0 \ldots\ldots 0 \\ 0 & 0 & 0 & 0 & 0 \ldots\ldots 0 \ldots\ldots 0 \\ 0 & 1 & 0 & 0 & 0 \ldots\ldots 0 \ldots\ldots 0 \\ 0 & 0 & 0 & 0 & 0 \ldots\ldots 0 \ldots\ldots 0 \end{array} \right| \begin{array}{l} \\ \left.\begin{array}{l} \\ \\ \end{array}\right\} 2 \\ \left.\begin{array}{l} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array}\right\} 2n+3 \\ \left.\begin{array}{l} \\ \\ \\ \\ \\ \end{array}\right\} 5 \end{array}$$

Proof: Follows from the lemmata 12–15. □

The depth of the linear cascade will be at most $(p + 1)$, where p is the number of product terms in the ESOP expression. The test sequence T can be stored in a ROM of size $= (2n + 10) \times (n + 3)$ bits. Table 2 shows the length of the test sequence for detecting single stuck-open faults in our GRM and ESOP implementation of some benchmark circuits (LGSynth '93 and Espresso). The test sequence is universal and robust. In most of the cases, the length of the test sequence is found to be even much smaller than the size of the test set for detecting stuck-at faults in conventional multilevel circuit implementation. For example, SIS generates 961 tests for *alu4* and 1245 tests for *apex5* to detect single stuck-at faults [19].

Table 2: Length of test sequence for detecting single stuck-open faults

Circuit	Inputs/outputs	Length of test sequence GRM	ESOP
9symml	9/1	49	28
adr4	8/5	45	26
alu4	14/8	69	38
apex5	117/88	481	244
ex4	128/28	525	266
mux	21/1	97	52
x1	51/35	217	112
x4	94/71	389	198
x2dn	82/56	31	174
x9dn	27/7	121	64

4 Conclusions

We have proposed easily testable CMOS implementations of both GRM and ESOP expressions for detecting all single stuck-open faults. Our design for an n-variable GRM expression needs a test sequence of length $(4n + 13)$. For ESOP a test sequence of length $(2n + 10)$ is sufficient. For each of these two cases, the test sequence is robust (independent of circuit delays) and universal (independent of the function being realised and structure of the CUT). The test sequences, being short in length, may be stored in a ROM on-chip for built-in self-testing. Thus, the control inputs used in the design will not increase the number of external I/O pins. For designing multioutput circuits, the NAND part and the EXOR tree may be appropriately shared and the same universal test sequence can then be used to test all single stuck-open faults. Logic optimisation for GRM and ESOP multioutput circuits needs further investigation.

5 References

1 Aitken, R.C.: 'Nanometer technology effects on fault models for IC testing', *IEEE Trans. Comput.*, 1999, **48**, pp. 46–51
2 Lee, C.H., Shen, K.H., Ku, T.K., Luo, C.H., Tso, C.C., Chou, H.W., and Hsia, C.: 'CVD Cu technology development for advanced Cu interconnect applications'. Proc. Conf. on Interconnect Technology, 2000, pp. 242–244
3 Needham, W., Prunty, C., and Yeoh, E.H.: 'High volume test escapes - An analysis of defects our tests are missing'. Proc. Int. Test Conf., 1999, pp. 25–34
4 Kundu, S., and Reddy, S.M.: 'Robust test for parity trees'. Proc. Int. Test Conf., 1988, pp. 680–687
5 Reddy, S.M., and Reddy, M.K.: 'Testable realization on FET stuck-open faults in CMOS combinational logic circuits', *IEEE Trans. Comput.*, 1986, **35**, (8), pp. 742–754
6 Kundu, S., Reddy, S.M., and Jha, N.K.: 'On the design of robust multiple-fault testable CMOS combinational logic circuits'. Proc. Int. Conf. on Computer-aided Design, Santa Clara, CA, 1988, pp. 240–243
7 Chakravarty, S.: 'On the complexity of computing test sets for complex CMOS gates', *IEEE Trans. Comput.-Aided Des.*, 1989, **8**, pp. 973–980
8 Kundu, S.: 'Design of multioutput CMOS combinational logic circuits for robust testability', *IEEE Trans. Comput.-Aided Des.*, 1989, **8**, pp. 1222–1226
9 Reddy, S.M., Reddy, M.K., and Kuhl, J.G.: 'On testable design for CMOS logic circuits'. Proc. Int. Test Conf., Philadelphia, PA, 1983, pp. 435–445
10 Liu, D.L., and McCluskey, E.J.: 'Designing CMOS circuits for testability', *IEEE Des. Test Comput.*, 1987, **4**, pp. 42–49
11 Jayasumana, A.P., Malaiya, Y.K., and Rajsuman, R.: 'Design of CMOS circuits for stuck-open testability', *IEEE J. Solid State Circuits*, 1991, **26**, pp. 58–61
12 Jha, N.K.: 'Multiple stuck-open fault detection in CMOS logic circuits', *IEEE Trans. Comput.*, 1988, **37**, pp. 426–432
13 Kundu, S., and Reddy, S.M.: 'On the design of robust testable CMOS

combinational logic circuits'. Proc. Int. Symp. on Fault-tolerant Computing, Japan, 1988, pp. 220–225

14 Das, D.K., Chakraborty, S., and Bhattacharya, B.B.: 'Universal and robust testing of stuck-open faults in Reed–Muller canonical CMOS circuits', *Int. J. Electron.*, 2003, **90**, (1), pp. 1–11

15 Sarabi, A., and Perkowski, M.A.: 'Fast exact and quasiminimal minimization of highly testable fixed-polarity AND/XOR canonical networks'. Proc. 29th Conf. on Design Automation, 1992, pp. 30–35

16 Wu, H., Perkowski, M.A., Zeng, X., and Zhuang, N.: 'Generalized partially-mixed-polarity Reed–Muller expansion and its fast computation', *IEEE Trans. Comput.*, 1996, **45**, pp. 1084–1088

17 Reddy, S.M.: 'Easily testable realizations for logic functions', *IEEE Trans. Comput.*, 1972, **21**, (11), pp. 1183–1188

18 Pradhan, D.K.: 'Universal test sets for multiple fault detection in AND–EXOR arrays', *IEEE Trans. Comput.*, 1978, **27**, (2), pp. 181–187

19 Kalay, U., Hall, D.V., and Perkowski, M.A.: 'A minimal universal test set for self-test of EXOR sum-of-products circuits', *IEEE Trans. Comput.*, 2000, **49**, (3), pp. 267–276

20 Saluja, K.K., and Reddy, S.M.: 'Fault detecting test sets for Reed–Muller canonic networks', *IEEE Trans. Comput.*, 1975, **24**, pp. 995–998

21 Bhattacharya, B.B., Gupta, B., Sarkar, S., and Choudhury, A.K.: 'Testable design of RMC networks with universal tests for detecting stuck-at and bridging faults', *IEE Proc. E, Comput. Digit. Tech.*, 1985, **132**, pp. 155–161

22 Sarabi, A., and Perkowski, M.A.: 'Design for testability properties of AND-EXOR networks'. Proc. Int. Workshop on Applications of the Reed-Muller Expansion in Circuit Design, 1993, pp. 418–424

23 Jabir, A., and Saul, J.: 'Heuristic AND-OR-EXOR three-level minimisation algorithm for multiple-output incompletely-specified boolean functions', *IEE Proc., Comput. Digit. Tech.*, 2000, **147**, (6), pp. 451–461

24 Song, N., and Perkowski, M.A.: 'Minimization of exclusive sum-of-products expressions for multiple-valued input, incompletely specified functions', *IEEE Trans. Comput.-Aided Des.*, 1996, **15**, pp. 285–395

25 Liu, P.K., and Mujio, J.C.: 'Boolean matrix transforms for the minimization of modulo-2 canonical expansions', *IEEE Trans. Comput.*, 1992, **44**, pp. 1012–1920

26 Purwar, S.: 'An efficient method of computing generalized Reed-Muller expansions from binary decision diagram', *IEEE Trans. Comput.*, 1991, **40**, pp. 1298–1301

27 McKenzie, L., Almaini, A.E., Miller, J.F., and Tompson, P.: 'Optimisation of Reed–Muller logic functions', *Int. J. Electron.*, 1993, **75**, pp. 451–466

28 Zillic, Z., and Vranesic, Z.: 'A multiple-valued Reed–Muller transform for incompletely specified functions', *IEEE Trans. Comput.*, 1995, **44**, (8), pp. 1012–1020

29 Song, N.: 'Minimization of exclusive sum-of-products expressions for multiple-valued input, incompletely specified functions'. MS thesis EE Dept., Portland State University, OR, 1992

30 Slusarczyk, A., Jozwiak, L., and Perkowski, M.A.: 'Term trees in application to an effective ATPG for AND-EXOR and AND-OR circuits'. Proc. Int. Workshop on the Applications of the Reed–Muller expansion in circuit design, Aug. 1999

31 Saul, J., Escherman, B., and Froessl, J.: 'Two-level logic circuits using EXOR sums-of-products', *IEE Proc. E, Comput. Digit. Tech.*, 1993, **140**, pp. 348–356

32 Damarala, T.R., and Karpovsky, M.: 'Fault detection in combinational networks by Reed–Muller transforms', *IEEE Trans. Comput.*, 1989, **38**, pp. 788–797

33 Debnath, D., and Sasao, T.: 'GRMIN: A heuristic simplification algorithm for generalised Reed–Muller expressions', *IEE Proc. Comput. Digit. Tech.*, 1996, **143**, (6), pp. 376–384

34 Drechsler, R., Hengster, H., Schfer, H., Hartmann, J., and Becker, B.: 'Testability of 2-level AND/EXOR circuits', *J. Electron. Test Theory Appl.*, 1999, **14**, pp. 219–225

35 Sasao, T.: 'Logic synthesis and optimization' (Kluwer Academic, Norwell, MA, 1993)

36 Sasao, T.: 'Easily testable realizations for generalized Reed–Muller expressions', *IEEE Trans. Comput.*, 1997, **46**, (6), pp. 709–716

37 Zhongliang, P.: 'Bridging fault detection for testable realization of logic functions'. Proc. Int. Conf. on VLSI Design, 2003, pp. 423–427

38 Wadsack, R.L.: 'Fault modeling and logic simulation of CMOS and MOS integrated circuits', *Bell Syst. Tech., J.*, 1978, **57**, pp. 1449–1474