Design For Testability Properties of AND/XOR Networks

Andisheh Sarabi, Marek A. Perkowski
Department of Electrical Engineering
Portland State University
Portland, OR 97207, USA

Abstract

AND/XOR networks include all realizations of switching functions in two and multi-level which include AND, XOR, and inverters as basic building blocks. It has been shown that for many AND/XOR canonical networks and trees the test set to detect stuck-at-faults and bridging faults are independent of the function being realized. This paper provides a survey of these studies and introduces certain results on some other AND/XOR Networks. Here a scheme for reduction of tests on canonical mixed polarity AND/XOR networks is introduced which can be extended to other general mixed polarity networks. In addition the independence of test sets for Consistent Generalized Reed-Muller trees is presented.

1 Introduction

Easy Testability of AND/XOR forms is one of the most important arguments for their use in practical design. Testability of XOR gates has long been known. However, AND/XOR functions have the special property that their required test set is independent of the actual function being realized. This property is of major importance in the design process. Among all the AND/XOR forms, it is the *CGRM* that has the least number of tests required.

With the advent of VLSI technology and the reduction in the cost of integrated circuit manufacturing, the testing aspects of the circuits have shown to take ever larger proportion of time and efforts. In recent years, there has been a shift to what is termed *Design and test* due to this phenomenon. While in classical design the process of design and testing of the integrated circuits were separated entities, in the new approaches to design, the testing aspect is integrated with the design process from the very initial stages and encompasses all levels from chip to board level. As problem of test generation is NP-complete [1], for large functions this becomes quite a formidable task. Having a universal set of tests for any function obviously reduces this problem drastically. Realization of switching functions in AND/XOR forms is exactly the case which results in an independent test set. This property will be presented in this paper. Before presenting the testability characteristics of these forms, certain terms and concepts related to tests and testing will be provided.

The basic aim of testing at the chip level is the identification of faults in a circuit. A em fault of a circuit is a physical defect of one or more components. The basic faults in any VLSI integrated circuit can be classified into two major categories, that of *perimetric faults* and *logical faults*. While the perimetric faults are related to the physical defects, the logical faults are associated with the logical aspects of the circuits. Perimetric faults are those responsible for alterations in the magnitudes of a circuit parameter, causing a change in some factor such as the circuit speed, current, or voltage. Logical faults on the other hand, are the ones that cause changes in the logic function of a circuit element or an input signal to some other logic function. The perimetric faults are of concern in the structural level design of logic circuits and the logical faults are associated with the functional level and this is precisely where the properties of the AND/XOR forms are of paramount importance. Other faults include those of *delay faults*, e.g. slow gates which usually only affect the timing performance and could result in hazards or critical races, and *intermittent faults* which occur only in some intervals and are very difficult to detect.

The logical faults in their place can be also divided into two classical classes of *bridging faults* and *stuck-at-faults*. Bridging faults occur particularly in MOS LSI circuits. These are mainly due to a short connection in the circuit between two or more lines resulting in circuit malfunction. This short circuit can be modeled as either a wired-AND or wired-OR function. The stuck-at-faults are themselves divided into stuck-at-1 and stuck-at-0 which occur at the inputs of the logic gates and cause the inputs to remain either at 1 or 0 permanently.

Design for testability is concerned with the addition of enough extra circuitry to a circuit or chip to reduce the complexity of its testing. Because of the decrease in the cost of hardware, as the amount of testing required rises, the design for testability techniques find a growing interest with VLSI circuit designers. These techniques are specially concerned with controllability and observability of internal logic values. *Controllability* is defined as a measure of how easily the internal logic of the circuit can be controlled from its primary inputs. *Observability*, on the other hand, is defined as a measure of how easily the internal logic of the circuit

[0]†This research was partially supported by the NSF grant MIP-9110772.

can be observed at its primary outputs. The process of test generation consists of the tasks of controlling and observing internal logic values.

AND/XOR forms, due to the testability characteristics of the XOR, have major properties of interest in design for testability. The small number of test sets and their independence of the switching function itself, as well as their controllability and observability are the major factors. This is due to the property of XOR that any change over one of its inputs is directly reflected on its output. Among all of these forms, Reed-Muller canonical forms require the least number of test sets for detecting stuck-at and bridging faults. *CGRM* forms have also the same number because the input variables retain the same polarity throughout the expansion. The other forms will be shown [2] to require larger test sets. In the following, the testability properties of the *RMC*, *CGRM*, and *CRMP* forms will be presented. This will be next contrasted with other forms. The presentation will be divided into discussion of stuck-at-faults and bridging faults. Each of these faults will also be divided into further subdivisions.

2 Background

While there are only two canonical forms for the Boolean AND/OR networks, the number of possible canonical representations of a function for the AND/XOR networks is very large [3]. One class of AND/XOR networks, the first to be noted in the literature, is that of the *Reed-Muller canonical (RMC)* form [4, 5] and its superset, the *Consistent Generalized Reed-Muller (CGRM)* canonical forms [3].

The *RMC* representation consists of only positive product terms and is given as:

$$f(x_1, x_2, \ldots, x_n) = \bigoplus_{i=0}^{2^n-1} a_i \mu_i \tag{1}$$

where $a_i \in \{0,1\}$ and $\mu_i = {x_n}^{e_n} {x_{n-1}}^{e_{n-1}} \ldots {x_2}^{e_2} {x_1}^{e_1} = \prod_{j=1}^{n} {x_j}^{e_j}$ where $e_j \in \{0,1\}$ such that $e_n e_{n-1} \ldots e_2 e_1$ is a binary number which equals i. Moreover ${x_i}^0 = 1$ and ${x_i}^1 = x_i$. $\oplus$ denotes the summ ation over GF(2), the Galois field of two elements.

If the restriction that all the variables in the function should take positive polarity is removed and they are also allowed to take negative polarities, one can have a *Canonical Restricted Mixed Polarity (CRMP)* form also known as *Generalized Reed-Muller (GRM)* canonical form. If the variables are, however, restricted to retain the same polarity, either positive or negative, in all product terms, the canonical form will be that of *Consistent Generalized Reed-Muller (CGRM)* form. This form is shown below:

$$f(x_1, x_2, \ldots, x_n) = \bigoplus_{i=0}^{2^n-1} a_i \dot{\mu}_i \tag{2}$$

where $a_i \in \{0,1\}$ and $\dot{\mu}_i = \dot{x}_n^{e_n} \dot{x}_{n-1}^{e_{n-1}} \ldots \dot{x}_2^{e_2} \dot{x}_1^{e_1} = \prod_{j=1}^{n} \dot{x}_j^{e_j}$ where $e_j \in \{0,1\}$ such that $e_n e_{n-1} \ldots e_2 e_1$ is a binary number which equals i; $\dot{x}_i^0 = 1$ and $\dot{x}_i^1 = \dot{x}_i$. $\dot{x}_i$ stands for x_i or $\bar{x}_i$ but not both. $\oplus$ again denotes summation over GF(2).

For n variables, there are 2^n possible arrangements of polarities; hence, there are 2^n possible *CGRM* forms. This number for *CRMP* forms is $2^{n2^{n-1}}$.

The largest class of AND/XOR networks is that of *Exclusive Sum of Products (ESOP)*. In this representation there is no restriction on the variables or product terms. This representation, although provides the the most economical representation of the function, it requires a test set of higher cardinality compared to the previous AND/XOR canonical forms.

Example 1 $1 \oplus x_1 \oplus \bar{x}_2 \oplus \bar{x}_1 x_2$ *is a* CRMP *form, because there exists only one term for each subset of variables. It is not a em CGRM form because x_1 appears both in a negative and a positive polarity.* $x_{\bar{x}_2} \oplus \bar{x}_1 x_2 \oplus x_2 \oplus x_1 x_2$ *is an* ESOP *which is not in a* CRMP *form because the term $x_1 x_2$ occurs more than once (in different polarities).* $\bar{x}_2 \oplus \bar{x}_1 \bar{x}_2$ *is a* CGRM *form, since both variables occur in constant polarities in the entire form. This is not a* RMC *form because variables are not positive. It is called a negative Reed-Muller form.*

It is also possible to derive AND/XOR networks which are essentially multi-level. These networks which have a tree like structure are produced by using three basic decomposition schemes. These decomposition schemes are those of Shannon, and two Davio expansions given below:

$$f = x_i \cdot f_{x_i} \oplus \bar{x}_i \cdot f_{\bar{x}_i} \tag{3}$$

$$f = f_{\bar{x}_i} \oplus x_i \cdot [f_{x_i} \oplus f_{\bar{x}_i}] \tag{4}$$

$$f = f_{x_i} \oplus \bar{x}_i \cdot [f_{x_i} \oplus f_{\bar{x}_i}] \tag{5}$$

where $f_{x_i} = f(x_1, \ldots, x_i = 1, \ldots, x_n)$ *and* $f_{\bar{x}_i} = f(x_1, \ldots, x_i = 0, \ldots, x_n)$.

Several AND/XOR tree networks can be generated by application of the above rules in different order of variables and combinations of the rules. If only rule (4) is applied for some fixed order of variables, the result will be a *Reed-Muller tree.* If for each variable either rule (4) or (5) are used, the result will be then a *Consistent Generalized Reed-Muller tree.* If either rule (3) or (4) or (5) are used for each of the variable, the result will be that of a *Kronecker Reed-Muller tree.* Still other tree networks are possible depending on the order of variables or the rule applied [6]. Flattening each of the trees into two levels will result in the corresponding canonical forms discussed before.

3 Detection of Stuck-at-Faults in CGRM Networks

The discussion of stuck-at-faults (SAF) can be presented in terms of single SAF and multiple SAF. Single SAF can further be investigated depending on whether the primary inputs are fault-free or not. These cases will be described in the following.

The first author to present the testability properties of Reed-Muller forms was Reddy [7]. The results obtained by Reddy for single SAFs are summarized in the following:

1) *"If the primary inputs leads are fault-free, then there exists a realization for an arbitrary $n-$variable logic function that requires a fault detection test set with only $n+4$ tests and this test is independent of the function being realized.*

2) *If the primary input leads could be faulty, then only $n+4+2n_e$ tests are required for detecting faults, where n_e is the number of variables appearing in an even number of terms in the Reed-Muller expansion for the function being realized.*

3) *If the primary input leads could be faulty, then by adding an extra observable output and an extra AND gate, the $(n+4)$ tests of 1) will be sufficient and these tests will again be independent of the function being realized."*

The circuit here is assumed to be composed of a cascade of XOR gates with a secondary input from AND gates each composed of different combinations of variables. An example of the this scheme for the function $f = 1 \oplus x_1x_2 \oplus x_1x_4 \oplus x_1x_2x_3 \oplus x_2x_3x_4 \oplus x_1x_2x_3x_4$is shown in Figure 1.

Figure 1 The cascade network for *RMC*

In order to detect single SAF in a cascade of XOR gates, it is sufficient to apply a set of tests covering all possible input combinations to each cell [7]. The following test set satisfies this purpose:

$$T_1 = \begin{bmatrix} a_0 & x_1 & x_2 & x_3 & . & . & . & x_n \\ 0 & 0 & 0 & 0 & . & . & . & 0 \\ 0 & 1 & 1 & 1 & . & . & . & 1 \\ 1 & 0 & 0 & 0 & . & . & . & 0 \\ 1 & 1 & 1 & 1 & . & . & . & 1 \end{bmatrix} \qquad (6)$$

Any stuck-at-0 fault at any AND gate input or output can be detected by applying either of the test inputs [0111...1, 1111...1]. Similarly, any stuck-at-1 fault at output of any AND gate can be detected by applying either of the test inputs [0000...0, 1000...0]. A stuck-at-1 fault at any input of the AND gates can be detected by the set T_2 below:

$$T_2 = \begin{bmatrix} d & 0 & 1 & 1 & . & . & . & 1 \\ d & 1 & 0 & 1 & . & . & . & 1 \\ d & 1 & 1 & 0 & . & . & . & 1 \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ d & 1 & 1 & 1 & . & . & . & 1 \end{bmatrix} \qquad (7)$$

where d stands for don't care. The total number of the tests in $T = T_1 \cup T_2$ is then the sum in T_1 and T_2 which is $4+n$.

The above results are true when none of the primary inputs are faulty. If any of the primary inputs are faulty as well, additional tests are required. As the XOR gate detects an odd number of changes, the $n+4$ tests above detect faults at those primary inputs that occur in an odd number of AND gates. For the primary inputs that occur in an even number of AND gates, as is suggested by Reddy, it is required to apply two tests for each of these inputs. In this method, the inputs of interest that occur in products with the smallest number of literals are chosen. The stuck-at-one is identified by choosing the inputs occurring in the products one at a time and assigning them zero along with the literals that do not occur in this product. The other variables in the product are assigned the value of 1. Detection of stuck-at-zeros is similar with the difference that the input variable of interest is assigned the value of 1 instead. These tests detect the stuck-at-0 and stuck-at-1 faults in the faulty primary inputs occurring in an even number of times. This leads to the point 2) above.

The idea behind point 3) is that having an additional AND gate with its inputs being the primary inputs that occur in an even number of products can reflect the faults at these primary inputs. The test set in point 1) will then be adequate to detect all the SAFs. In order to make sure the faults detected are not due to the primary inputs, another AND gate can be added exactly the same as above. For further discussion the reader can refer to [7].

The test sets given by Reddy above were shown by Kodandapani [8] to be reducible. Kodandapani has shown that by assigning specific values to the don't cares in the matrix T_2, and a certain scheme of reorganizing the terms, one of the tests in T_1 can be reduced. In this way, the number of tests required to detect any single stuck-at-fault in an AND gate or a single faulty XOR gate can be reduced to $n+3$.

The above results were for the cases where only single SAFs are involved. When there are multiple faults involved, the test set is again shown to be independent of the function. Saluja and Reddy [9] have shown that to detect t faults, $t \geq 1$, only

$$4 + \sum_{i=1}^{\lfloor log_2 2t \rfloor} \binom{n}{i} \tag{8}$$

tests are required to detect all t-multiple stuck-at-faults (where $\lfloor x \rfloor$ stands for the integer part of x). With addition of an extra AND gate and one observable output, the single stuck-at-faults can be detected as well. Furthermore, Saluja has shown that by addition of extra observable outputs, the same $n+4$ independent tests can detect all single and multiple stuck-at-faults [10].

It can be easily shown that any universal test set generated for detection of single stuck-at-faults of an *RMC* form can be modified for any *CGRM* form by just inverting the test bits for those variables which are of negative polarity in the *CGRM*. As the input to the AND gates are the complements of the test bits for the case of complemented variables, this inversion makes the test bit to be equivalent to the *RMC* network described above. Hence the same results hold true for the *CGRM* networks. This is shown by an example below:

Example 2 *Let a network be represented by* $1 \oplus x_1x_2 \oplus x_1x_4 \oplus x_1x_2x_3 \oplus x_2x_3x_4 \oplus x_1x_2x_3x_4$. *This is an* RMC *network with the following universal tests for detection of the stuck-at and bridging faults:*

$$T_1 = \begin{bmatrix} a_0 & x_1 & x_2 & x_3 & x_4 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{9}$$

$$T_2 = \begin{bmatrix} d & 0 & 1 & 1 & 1 \\ d & 1 & 0 & 1 & 1 \\ d & 1 & 1 & 0 & 1 \\ d & 1 & 1 & 1 & 0 \end{bmatrix} \tag{10}$$

Now let us assume a CGRM *network has been given in the form* $1 \oplus \bar{x}_1x_2 \oplus \bar{x}_1\bar{x}_4 \oplus \bar{x}_1x_2x_3 \oplus x_2x_3\bar{x}_4 \oplus \bar{x}_1x_2x_3\bar{x}_4$. *In this form,* x_1 *and* x_4 *have negative polarities and thus their respective columns are complemented with respect to the above* RMC *network. The universal tests for detection of the stuck-at and bridging faults of the* CGRM *network will be:*

$$T_1 = \begin{bmatrix} a_0 & x_1 & x_2 & x_3 & x_4 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix} \tag{11}$$

$$T_2 = \begin{bmatrix} d & 1 & 1 & 1 & 0 \\ d & 0 & 0 & 1 & 0 \\ d & 0 & 1 & 0 & 0 \\ d & 0 & 1 & 1 & 1 \end{bmatrix} \square \tag{12}$$

If the inversions are to be generated internally, an extra AND gate with observable output can be added to the network. This AND gate should have as input all the variables that appear with negative polarity in the *CGRM* form. This gate will detect the faults produced by any of the inverters.

It has to be noted that the same tests would detect the stuck-at-faults of a multi-output *CGRM* network. The difference is that the observable points will be the same as the outputs and they have to be examined for each of the functions being realized.

4 Detection of Bridging Faults of CGRM Networks

The *RMC* networks not only have universal test sets for detection of stuck-at-faults, but there exist similar schemes for detection of the bridging faults. It has been shown [11] that with certain modifications the same universal tests for detection of stuck-at-faults can be utilized to detect bridging faults of the *RMC* networks.

Before describing the bridging fault detection schemes, certain classification of the bridging faults will be described. Bridging faults result from short connections in the circuits. These shorts can occur at the inputs of logic gates, input lines to different logic gates, or between the lines of the same logic level. These bridging faults are called *intragate*, *intergate*, and *intralevel* respectively. The bridging faults can in turn be either wired-AND or wired-OR function, depending on positive or negative logic.

The results obtained by Bhattacharya, et al regarding the bridging faults detection of *RMC* networks are as follows:

Theorem 1 [Bhattacharya] *An* RMC *network of* n *variable function can be augmented by adding an extra AND gate, with all input variables as its input, so that the universal test set* T, *of cardinality* $n+4$ *is sufficient to detect the different intralevel OR-bridging faults, as well as all single stuck-at-faults.*

This augmentation for the network in Figure 1 is shown in Figure 2.

Theorem 2 [Bhattacharya] *An* RMC *network of* n *variable function can be augmented by adding an extra OR gate so that the universal test set* T_u, *of cardinality* $2n+4$ *is sufficient to detect the different intralevel AND-bridging faults, as well as all single stuck-at-faults.* $T_u = T \cup T_\alpha$ *where*

$$T_\alpha = \begin{bmatrix} a_0 & x_1 & x_2 & x_3 & . & . & . & x_n \\ d & 1 & 0 & 0 & . & . & . & 0 \\ d & 0 & 1 & 0 & . & . & . & 0 \\ d & 0 & 0 & 1 & . & . & . & 0 \\ d & 0 & 0 & 0 & . & . & . & 0 \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ d & 0 & 0 & 0 & . & . & . & 1 \end{bmatrix} \tag{13}$$

The scheme for addition of the OR gate is similar to the one shown in Figure 2 with the difference that the augmented AND gate is replaced with an OR gate of all input variables.

For the case of *CGRM* networks, Bhattacharya, et al propose a different circuit augmentation scheme. In this case for detection of OR-bridging faults, it is proposed that three additional gates be added to the network. These gates are one n-input AND gate of all primary inputs, one n-input OR gate of all primary inputs, and an n_1-input AND gate, where n_1 is the number of complemented literals in the *CGRM* expansion. The inputs to this AND gate are derived from the outputs of the inverters producing the negated inputs. The test set devised for the *RMC* network to detect AND-bridging faults can then be applied to detect the OR-bridging faults of the *CGRM* network. A similar augmentation can be devised for detection of the AND-bridging faults for the *CGRM* network.

Figure 2 Augmentation of the network in Figure 1 for detection of OR-bridging faults

If no augmentation is to be incorporated, Damarla and Karpovsky [12] give an upper bound for the number of test patterns to detect all single Stuck-at-faults and all single detectable AND and OR bridging faults of an *RMC* network. In this scheme for an *RMC* network with k outputs and n inputs ($k \leq 2^n$), at most $3n + 5$ test patterns are needed to detect all single stuck-at-faults and both AND and OR bridging faults which are detectable.

5 Detection of Stuck-at-Faults in Mixed-Polarity Networks

Pradhan [2] has given a universal test set for multiple fault detection of mixed polarity AND/XOR networks when the inversion of inputs is produced internally using XOR gates. This universal test set which is independent of the function was shown to be of cardinality

$$6 + 2n + \sum_{e=0}^{j} \binom{n}{e} \tag{14}$$

where n is the number of variables of order j, the maximum number of literals contained in any product term in the AND/XOR expression. This universal test set is comprised of T_3, T_4, and T_5, where

$$T_{31} = \begin{bmatrix} a_0 & x_1 & x_2 & x_3 & . & . & . & x_n \\ 1 & 0 & 0 & 0 & . & . & . & 0 \\ 0 & 1 & 1 & 1 & . & . & . & 1 \end{bmatrix} \tag{15}$$

$$T_{32} = \begin{bmatrix} 1 & 1 & 0 & 0 & . & . & . & 0 \\ 1 & 0 & 1 & 0 & . & . & . & 0 \\ 1 & 0 & 0 & 1 & . & . & . & 0 \\ 1 & 0 & 0 & 0 & . & . & . & 0 \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ 1 & 0 & 0 & 0 & . & . & . & 1 \end{bmatrix} \tag{16}$$

$$T_{33} = \begin{bmatrix} 0 & 0 & 1 & 1 & . & . & . & 1 \\ 0 & 1 & 0 & 1 & . & . & . & 1 \\ 0 & 1 & 1 & 0 & . & . & . & 1 \\ 0 & 1 & 0 & 1 & . & . & . & 1 \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ 0 & 1 & 1 & 1 & . & . & . & 0 \end{bmatrix} \tag{17}$$

$$T_4 = \begin{bmatrix} 0 & 0 & 0 & 0 & . & . & . & 0 \\ 1 & 0 & 0 & 0 & . & . & . & 0 \\ 0 & 1 & 1 & 1 & . & . & . & 1 \\ 1 & 0 & 0 & 0 & . & . & . & 0 \end{bmatrix} \tag{18}$$

and $T_5 = T_2^j$, where T_2^j is the set of n-vectors with the number of 1's in the vectors being less than or equal to j.

6 Detection of Stuck-at and Bridging Faults in CRMP Networks

The Canonical Restricted Mixed Polarity AND/XOR networks are the largest class of AND/XOR canonical networks where a product term of certain literals occurs only once in the network. Moreover, a *CRMP* network can be decomposed into its component *CGRM* networks. Each *CGRM* network can be then examined with its universal test sets for detection of stuck-at and bridging faults using the procedures described in previous sections. In this way, it is possible to devise a method for the case of *CRMP* networks which can result in a reduced number of tests as compared to the result obtained by Pradhan in the general case.

In [13] a method for minimization of functions in *CRMP* form was presented. In this method a quasi-minimal representation of function based on the number of terms was given. In this minimal form, it was shown that all terms are subcombinations of the prime terms of the function, where a prime term is a term in which the boolean difference of the function with respect to that term is identically equal to 1. In this form, certain terms include literals of the same polarity and hence can be grouped together as component *CGRM*s of that form. For r component *CGRM* networks, the number of tests required to detect stuck-at and bridging faults can be r times the number of test sets for each component, provided that there are r observable outputs corresponding to the component *CGRM*s. However, as some of the variables occur in different polarities in different component *CGRM* networks, there will be some tests which would occur more than once. There will also exist tests which are termed *compatible*. By combining the compatible tests and applying the repeated tests only once, it is possible to reduce the overall number of tests.

Definition 1 *Two tests are* compatible *if all corresponding bits in the two test vectors are compatible. Two test bits are* compatible *if one of them is don't care or they are both the same.*

Example 3 *10-0100 and 1-0-1-0 are compatible while 10-0100 and 1010000 are not.* □

Withidentification of the compatible tests, it is possible to construct a compatibility graph. In this graph the nodes represent the tests and the nodes that are compatible will be adjacent. The problem of reducing the number of tests for a *CRMP* network can then be formulated as follows:

- Decompose the *CRMP* network into its component *CGRM* networks.
- Generate the test sets for each of the component *CGRM* networks.
- Remove the repeated occurrences of the same tests.
- Construct the compatibility graph of the tests.
- Find the disjoint covering of the graph with maximum cliques.

Once the covering is chosen, a single test is created for each group of compatible tests by combining them (0 and d gives 0, 1 and d gives 1). As an example, the test used for the two compatible tests d10 and 1dd will be 110.

The test reduction method will be shown by Example 4 below:

Example 4 *Let a* CRMP *network be represented by* $1 \oplus \bar{x}_1x_2 \oplus x_1\bar{x}_4 \oplus x_1x_2x_3 \oplus \bar{x}_2x_3\bar{x}_4 \oplus \bar{x}_1x_2x_3x_4$*. This network can clearly be represented by 3 component* CGRM *networks given by:* $C_1 = 1 \oplus \bar{x}_1x_2 \oplus \bar{x}_1x_2x_3x_4$, $C_2 = x_1\bar{x}_4 \oplus \bar{x}_2x_3\bar{x}_4$, *and* $C_3 = x_1x_2x_3$. *The test sets for each of the component* CGRM *networks are then:*

$$C_1 : \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ d & 1 & 1 & 1 & 1 \\ d & 0 & 0 & 1 & 1 \\ d & 0 & 1 & 0 & 1 \\ d & 0 & 1 & 1 & 0 \end{bmatrix} \quad C_2 : \begin{bmatrix} d & 0 & 1 & 0 & 1 \\ d & 1 & 0 & 1 & 0 \\ d & 0 & 0 & 1 & 0 \\ d & 1 & 1 & 1 & 0 \\ d & 1 & 0 & 0 & 0 \\ d & 1 & 0 & 1 & 1 \end{bmatrix} \tag{19}$$

$$C_3 : \begin{bmatrix} d & 0 & 0 & 0 & d \\ d & 1 & 1 & 1 & d \\ d & 0 & 1 & 1 & d \\ d & 1 & 0 & 1 & d \\ d & 1 & 1 & 0 & d \\ d & 1 & 1 & 1 & d \end{bmatrix} \tag{20}$$

It can be seen that d111d in C_3 *occurs twice and one of them will be used. It is the same case with d0101 in* C_1 *and the one in* C_2*. The compatible tests are 01000, d1000, 00111, d011d, 11000, d1000, 10111, d011d, d1111, d111d, d0110, d011d, d1010, d101d, d1110, d111d, and d1011, d101d while d0011, d0101, d0010, and d110d are not compatible with any other tests. Now, a minimal number of non-redundant tests can be devised where only one test will be created from each set of compatible tests.* □

As it can be seen, in this method the test sets would no longer be universal: however, depending on the number of component *CGRM* networks, the variables present and their polarities, it may be possible to reduce the number of tests drastically. The merits of the method vary with the type of the *CRMP* network and will not always yield the same gains. It also requires as many observable points as the number of component *CGRM* networks.

7 Detection of Stuck-at-Faults in Reed-Muller Trees

The basic tree structure realizing the Reed-Muller trees is shown in Figure 3. In this structure, the inputs to any AND gate is one of the input variables and the output of one XOR gate from a previous level. The inputs to an XOR gate is either an AND gate or another XOR gate. The tree expansion terminates with 0 or 1 at certain branches. In this case, the constants will be the input to one of the XOR gates and that branch will stop from expanding.

Let us observe that applying 1s to the input variables will result in all the AND gates to become transparent; i.e. the tree will be one of XOR-tree. It has been shown [14] that four tests are necessary and sufficient to detect all single stuck-at-faults of an XOR-tree. Moreover, it has been shown [15] that at most $\lfloor 3n/2 \rfloor + 1$ tests would detect all multiple-stuck-at faults of XOR-trees. To detect stuck-at-faults of the AND gates the following scheme is applied. It can be seen that applying 0s for all the constant inputs to the XOR gates and 1s for all input variables and constant inputs to the AND gates will detect stuck-at-0s at the

AND input gates. Detecting stuck-at-1 faults at the inputs to AND gates requires a set of $n+1$ tests shown below:

$$
T_r = \begin{array}{c}
\begin{array}{cccccccc} a_1 & \cdots & a_m & a_p & x_1 & x_2 & \cdots & x_n \end{array} \\
\begin{bmatrix}
1 & \cdots & 0 & 1 & 1 & \cdot & \cdots & 1 \\
1 & \cdots & 1 & 0 & 1 & \cdot & \cdots & 1 \\
1 & \cdots & 1 & 1 & 0 & \cdot & \cdots & 1 \\
1 & \cdots & 1 & 1 & 1 & \cdot & \cdots & 1 \\
\cdot & \cdots & \cdot & \cdot & \cdot & \cdot & \cdots & \cdot \\
\cdot & \cdots & \cdot & \cdot & \cdot & \cdot & \cdots & \cdot \\
\cdot & \cdots & \cdot & \cdot & \cdot & \cdot & \cdots & \cdot \\
1 & \cdots & 1 & 1 & 1 & \cdot & \cdots & 0
\end{bmatrix}
\end{array} \tag{21}
$$

where a_1 through a_m represent the constant inputs to the XOR gates and a_p represents the constant to the last level AND gate.

Figure 3 The Reed-Muller Tree for a 3-variable function

Similar to the *CGRM* forms, for the case of *CGRM* trees, there are some input variables that will be in negative polarities but the structure is similar to the Reed-Muller tree. Here, for those variables that occur in negative polarity, the tests need to be the bit-wise opposite to the corresponding test bit in the Reed-Muller tree. If the inversions are to occur inside the chip, an extra AND gate with observable output having all the negated inputs as its input would be required.

8 Conclusions

In this paper the testability of different AND/XOR networks was given. As it is evident, universal test sets exist for the networks presented which are independent of the function being realized. In addition, a method for minimization of the number of tests required to detect faults for *CRMP* networks, as well as fault detection of Consistent Generalized Reed-Muller trees were introduced.

The minimized tests for *CRMP* networks can reduce the overall number of tests required as compared to the universal test set for general mixed-polarity AND/XOR networks. This method can be generalized to other AND/XOR networks where both positive and negative polarities of the same literals are involved.

The universal tests of the tree networks discussed also show the merits of AND/XOR networks in design for testability. It can be further investigated whether the bridging faults and tree networks with both polarities of literals also display the independence of tests set as it is true for the two level cases discussed.

These latter aspects will be further expanded in the final text.

References

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

[2] Pradhan, D. K. "Universal Test Sets for Multiple Fault Detection in AND-EXOR Arrays" *IEEE Transactions on Computers*, V. C-27, No. 2, February 1978.

[3] A Davio, M., Deschamps, J. P., and Thayse, A. "Discrete and Switching Functions" *Mc-Graw-Hill*, 1978.

[4] Reed, I. S. "A Class of Multiple-Error-Correcting Codes and Their Decoding Scheme" *IRE Transactions on Information Theory*, V. PGIT-4, pp. 38-49, 1954.

[5] Muller, D. E. "Application of Boolean Algebra to Switching Circuit Design and to Error Detection" *IRE Transactions on Electronic Computers*, V. EC-3, pp. 6-12, September 1954.

[6] Perkowski, M. A. "The Generalized Orthonormal Expansion of Functions With Multiple-Valued" Inputs and Some of its Applications *ISMVL'92*, pp. 442- 450, June 1992.

[7] Reddy, S. M. "Easily Testable Realization for Logic Functions" *IEEE Transactions on Computers*, V. C-21, No. 11, pp. 1183-1188, November 1972.

[8] Kodandapani, K. L. "A Note on Easily Testable Realizations for Logic Functions" *IEEE Transactions on Computers*, V. C-23, pp. 332-333, 1974.

[9] Saluja, K. K. and Reddy, S. M. "Fault Detecting Test Set for Reed-Muller Canonic Networks" *IEEE Transactions on Computers*, V. C-24, No. 10, pp. 995-998, 1975.

[10] Saluja, K. K. "A Study of Combinational Networks Based on Reed-Muller Canonic Forms" *Ph.D. Dissertation, University of Iowa, Iowa City*, 1973.

[11] 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 Proceedings*, V. 132, Part E, No. 3, pp. 155-161, May 1985.

[12] Damarla, T., Karpovsky, M. "Detection of Stuck-at and Bridging Faults and Bridging Faults in Reed-Muller Canonical (RMC) Networks" *IEE Proceedings*, V. 136, Part E, No. 5, pp. 430-433, September 1989.

[13] Perkowski, M. A., Csanky, L., Sarabi, A., and Sch:afer, I. "Fast Minimization of Mixed-Polarity AND/XOR Canonical Networks" *Proceedings of ICCD'92*, pp. 33-36, Cambridge, MA, October 1992.

[14] Hayes, J. P. "On Realization of Boolean Functions Requiring a Minimal or Near-Minimal Number of Tests" *IEEE Transactions on Computers*, V. C-20, No. 12, pp. 1506-1513, December 1971.

[15] A Seth, S. C. and Kodandapani, K. L. "Diagnosis of Faults in Linear Tree Networks" *IEEE Transactions on Computers*, V. C-26, No. 1, pp. 29-33, anuary 1977.