CHAPTER 3

MATERIALS AND METHODS

3.1 INTRODUCTION

This chapter explains about the materials and methods used in the research work. Also this chapter details about the generalized network structure, formation of test vectors, role of test vectors in fault detection, significance of control vectors and their assignments, computation of identifiability and distinguishability factors in decimal forms and finally the ten random functions considered for testability. The procedure for fault detection and diagnosis is explained with flow charts.

3.2 NETWORK STRUCTURE

Five network structures, each with minor modifications, for an easily testable Boolean circuit employing ESOP Reed-Muller canonical expression, required to be tested with a very limited number of input combinations have been used. They mainly comprise AND gates and XOR gates. The AND gates produce the product terms given in a function, while the XOR gates are used to produce the circuit function output. This set of XOR gates may be called as ‘function XOR block’.

Additional XOR gates may be necessary to produce the complemented variables appearing in the given function. These XOR gates can be called as ‘literal-complementing XOR gates’, as they produce the
complemented values of the data variables in the given function. A few control lines as well as auxiliary outputs are also included for better tractability of the faults. The connection of these control lines is also discussed.

Next, the test set or the set of input combinations required to test the testable realization is explained, incorporating the reasons for such a test set. The minimum cardinality of the test sets is just \( (n+5) \), where \( n \) is the number of data variables. It consists of three subsets, one for testing the XOR gate producing the primary output function, one for testing the AND gates and yet another for testing the literal-complementing XOR gates. Each of the test vectors comprise both the control inputs and the actual data inputs. The basic procedure of testing the testable circuit is also given.

![Diagram](image)

**Figure 3.1 Generalized network structure**

The basic network structure of the schemes is the same as that proposed by Zhongliang (2002) and is shown in Figure 3.1. The actual data inputs to the system are \( x_1, x_2, \ldots, x_n \). Additionally, the scheme requires four
control inputs $c_1$ to $c_4$. The literal-complementing block along with $c_1$ produces the complements of the literals used in the function. Only those literals appearing in a complemented form require an XOR gate in this block.

The literals of each product term are combined through an AND gate. The product terms in the given function are designated as $p_1$, $p_2$, ..., $p_m$. Hence, the number of AND gates required is equal to the number of product terms in the logic function. Further, each of the AND gates of this block has an additional input from one of the control lines $c_1$, $c_2$, $c_3$ and $c_4$ depending on the number of gates used in the XOR tree block producing the final function $F$. Finally, all the data and complementary gate outputs are applied to a separate AND gate and an OR gate, producing auxiliary outputs $O_1$ and $O_2$, to aid in the detection of faults which cannot be differentiated by the main function output $F$ alone.

### 3.3 TEST VECTORS

Zhongliang (2002) proposed a test matrix for the detection of single stuck-at faults. The total number of rows is $(n+5)$. Each of its rows is a $(n+4)$ long vector, $n$ being the number of data inputs. The first four columns of the matrix represent the control inputs $c_1$ to $c_4$ while the remaining $n$ columns are that of the data inputs $x_1$ to $x_n$. The basic idea behind the selection of such a test set is the testability requirements of the XOR tree, the AND array, as also the data and control inputs, forming the important components of the network structure. The testability of each of them is explained below.

#### 3.3.1 Fault Detection of XOR Tree

Let $v_1 = (0,0,1)$, $v_2 = (0,1,0)$ and $v_3 = (0,1,1)$. The following relations exist among the vectors, $v_1 = v_2 \oplus v_3$, $v_2 = v_3 \oplus v_1$ and $v_3 = v_1 \oplus v_2$. One of these vectors is assigned to the output of XOR tree, and the others to the
inputs of the XOR gate. This procedure is repeated until every primary input in the XOR tree has been assigned a vector from among \(v_1, v_2\) and \(v_3\). Combining the components that belong to the vectors assigned for the primary input of XOR tree, three m-tuple vectors that are able to detect any fault in XOR tree can be obtained. For example, the three vectors are \((0,0,0,0,0,0,0,0,0)\), \((1,0,0,1,0,1,1,1)\), and \((1,1,1,0,1,0,0,1)\) for the circuit in the Figure 3.2.

Four control input lines \(c_1, c_2, c_3\) and \(c_4\) are used to detect the faults of XOR tree. The method that \(c_i (2 \leq i \leq 4)\) connects to every AND gate in AND array is as given in the section 3.4.

Let set \(T_i = \{a_1, a_2, a_3\}\) where \(a_i, (1 \leq i \leq 3)\), is a vector of length \((n+4)\), comprising the control inputs \(c_1\) to \(c_4\) and the data inputs \(x_1\) to \(x_n\), in order and defined as in the Table 3.1.

### Table 3.1 Test set for XOR tree

<table>
<thead>
<tr>
<th>(T_i)</th>
<th>(c_1)</th>
<th>(c_2)</th>
<th>(c_3)</th>
<th>(c_4)</th>
<th>(x_1)</th>
<th>(x_2)</th>
<th>...</th>
<th>(x_n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a_1)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
</tr>
<tr>
<td>(a_2)</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>(a_3)</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>...</td>
<td>1</td>
</tr>
</tbody>
</table>

Although the complement of \(x_i\) may also exist in the logic function \(f\), the XOR gates in the input and control parts of ESOP circuit convert it into PPRM form when the control input \(c_1\) is set to 0. Therefore, the AND array is able to apply the above test vectors to the inputs of the XOR tree. Thus, \(T_i = \{a_1, a_2, a_3\}\) is the test set of XOR tree.
3.3.2 Fault Detection of AND Array

For an AND gate with \( s \) inputs, the single fault test set consists of \((s+1)\) vectors \( b_i \), \( (0 \leq i \leq s) \) as shown in the Table 3.2. All the components in the vectors \( b_0 \) are equal to 1; the \( i \)-th component in vector \( b_i \) for \( (1 \leq i \leq s) \) equals 0, while all the other components are 1. In order to detect the faults in AND array, the control input \( c_1 \) is set to 0, wherein the PPRM form is obtained, and a 0 or 1 to every input of any AND gate in AND array is applied by choosing some values for the primary inputs of ESOP circuit.

<table>
<thead>
<tr>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>…</th>
<th>( x_s )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>…</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>…</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>…</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>…</td>
<td>1</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>…</td>
<td>0</td>
</tr>
</tbody>
</table>

Let \( d_i \) \( (0 \leq i \leq n) \) be defined as a vector of \((n+4)\) components as in Table 3.3.

Since the XOR gate has the feature that the path that from one input line of the gate to primary output will be sensitized due to a change in the other input, the vector \( d_0 \) can detect all the \( s\)-\( a\)-0 faults in the AND array;
vector $d_i$ ($1 \leq i \leq n$) can detect all the s-a-1 faults. Thus, the set $T_2 = \{d_0, d_1, d_2, \ldots, d_n\}$ is a test set for the AND array.

**Table 3.3 Test set for AND gate array of figure 3.1**

<table>
<thead>
<tr>
<th></th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$c_4$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>\ldots</th>
<th>$x_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_0$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>\ldots</td>
<td>1</td>
</tr>
<tr>
<td>$d_1$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>\ldots</td>
<td>1</td>
</tr>
<tr>
<td>$d_2$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>\ldots</td>
<td>1</td>
</tr>
<tr>
<td>$d_3$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>\ldots</td>
<td>1</td>
</tr>
<tr>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
</tr>
<tr>
<td>$d_n$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>\ldots</td>
<td>0</td>
</tr>
</tbody>
</table>

### 3.3.3 Fault Detection of Data and Control Inputs

The test set $T_3$ for the input and control parts is given by $T_3 = \{e_1, e_2, e_3\}$, where $e_1$, $e_2$ and $e_3$ are defined as indicated in Table 3.4.

**Table 3.4 Test set for data and control inputs**

<table>
<thead>
<tr>
<th></th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$c_4$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>\ldots</th>
<th>$x_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_1$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>\ldots</td>
<td>1</td>
</tr>
<tr>
<td>$e_2$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>\ldots</td>
<td>0</td>
</tr>
<tr>
<td>$e_3$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>\ldots</td>
<td>0</td>
</tr>
</tbody>
</table>

All the s-a-0 faults in this circuit part, except for the s-a-0 fault of control input $c_1$, can be detected by the vector $e_1$ and the observable output.
O₁. All the s-a-1 faults can be detected by the vector e₂ and the observable output O₂. The s-a-0 fault of the control input c₁ can be detected by the vector e₃ and the observable output O₂.

3.3.4 Total Test Set for ESOP Circuits

The cardinality of sets T₁, T₂ and T₃ is 3, (n+1) and 3 respectively. There exists the same vector in T₁ and T₃, i.e. a₁ in T₁ and e₂ in T₃; also one vector is common to T₂ and T₃, viz. d₀ in T₂ and e₁ in T₃. Thus, the test set of ESOP circuit consists of only (n+5) vectors, where n is the number of variables in logic function F. Thus, the generalized test set pattern T is as shown in the Table 3.5.

<table>
<thead>
<tr>
<th></th>
<th>c₁</th>
<th>c₂</th>
<th>c₃</th>
<th>c₄</th>
<th>x₁</th>
<th>x₂</th>
<th>…</th>
<th>xₙ</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>…</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>…</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>…</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>…</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>…</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>…</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>…</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>…</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>…</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>…</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3.5 Complete generalized test set
The first test vector is an all-zero vector. The second vector has only \( c_1 \) and \( c_2 \) as zero, while in the third vector only \( c_1 \) and \( c_3 \) are zero valued. The next vector consists of all 1’s except for \( c_1 \). The next \( n \) vectors are made up of ‘walking zero vectors’ i.e. only a single zero is present in each of the vectors (ignoring the first element). This zero gets shifted by a column for each of the succeeding vectors. This produces an illusion of the zero ‘walking’ through the columns and hence the name. Finally, the last vector is an all-zero one except for the first element. The test matrix is the same for any logic function in Reed-Muller ESOP form for a given \( n \).

3.4 CONTROL SIGNAL ASSIGNMENT

The required control signals for each AND gate are determined as follows,

![Diagram of control signal assignment](image)

**Figure 3.2 Control input determination**
Step 1: Draw the XOR gate tree for the required product terms of the given function.

Step 2: Assign the numerals 1, 2 and 3 respectively to the two inputs and the output of the final XOR gate producing the function output F.

Step 3: Consider each XOR gate connected to the inputs of the final XOR gate considered in step 2. Assign the outputs of these XOR gates with the same numbers as the inputs of the final XOR gate of step 2.

Step 4: If the output of the XOR gate considered is 1, then assign 2 and 3 to its inputs; else if the output is numbered 2, assign 3 and 1 to its inputs.

Step 5: Now consider the next earlier input stage and assign the same numerals as the output points connected from the previous steps.

Step 6: Assign cyclic numerals as discussed in previous steps.

Step 7: Repeat steps 6 and 7 until the basic input stage of the XOR gate tree is reached.

Step 8: The numerals indexing this input stage incremented by 1 indicates the cardinality of the control line to be connected to the AND gate block.

The above procedure is shown as a flowchart in Figure 3.3.
Figure 3.3 Flowchart for control input determination
3.5 TEST PROCEDURE

**Step 1:** Set up the circuit as in Figure 3.1.

**Step 2:** Assign control line $c_1$ for complementing the variables as indicated in the Figure 3.1.

**Step 3:** Connect the control lines $c_2$ to $c_4$ as explained in the section 3.4.

**Step 4:** Apply the test vectors as given in the Table 3.5, one by one.

**Step 5:** For each test vector applied, determine the three fault-free outputs $F$, $O_1$ and $O_2$.

**Step 6:** Simulate the single stuck-at type of fault and determine the corresponding outputs $F$, $O_1$ and $O_2$.

**Step 7:** Compare the sets of outputs with the predetermined fault-free outputs.

**Step 8:** If the two output sets match exactly, it implies that a fault, if present, is not identifiable or detectable. Else, the fault is a detectable one.

**Step 9:** Repeat steps 4 to 8 for all inputs and outputs of all gates except those of auxiliary outputs.

**Step 10:** Repeat steps 4 to 8 for double stuck-at, OR-bridging and AND-bridging faults for other possible combination pairs of control inputs, data inputs and intermediate gate outputs in the network.

**Step 11:** For all the faults, the identifiability factor and distinguishability factor are calculated as explained in section 3.6.1.
The flow chart for the above procedure is shown in Figure 3.4.

![Flowchart of test procedure](image-url)
3.6 RESULT COMPACTION TECHNIQUE

There are more than one output for the circuits, namely F, O₁ and O₂ or F and O. Thus, two/three binary outputs are obtained for each test vector. Hence, the total size of the binary results due to the application of the test set for a fault (s-a-0 or s-a-1) in one line is $2 \times 3 \times$ number of test vectors for a three data circuit. Considering all the possible faults (even of same type say single stuck-at type) in different lines, both external and internal, the binary results may require a huge memory space. The result compaction process simply considers all the binary outputs of an output for the entire test set for each fault as a multi-bit binary number, which is converted into its decimal equivalent, assuming the output for the first input vector as the most significant bit. The decimal number so generated for each output and each fault case can be easily stored in a small number of memory locations. It is also obvious that the decimal results are convenient to tabulate than the binary form.

3.6.1 Computation of identifiability and distinguishability factors

The procedure for calculating the identifiability and distinguishability factors are as follows, namely,

**Step 1:** The set of Boolean outputs (main and auxiliary) for each test vector is determined as mentioned in the section 3.5 for the given circuit assuming it to be fault free.

**Step 2:** The output Boolean vectors are each considered separately and their decimal equivalents are determined by standard binary to decimal conversion process.
Step 3: The above two steps are repeated assuming or simulating single stuck-at faults at various locations of the circuit.

Step 4: The total number of faults for which the decimal equivalents of the outputs sets are identical to the output set of the fault-free operation is counted.

Step 5: The total number of possible faults is determined from a knowledge of input and output points including intermediate gates of the circuit.

Step 6: The ratio in percentage of the total number of faults obtained in step 4 to the total number of possible faults as known in step 5 gives a measure of the unidentifiable faults, from which the identifiability factor can be determined.

Step 7: The number of total faults for which the decimal equivalents of the output sets F, O₁ and O₂ are different from the fault-free output sets are counted.

Step 8: The ratio of the number counted in step 7 to the total number of possible faults is the indistinguishability index. The distinguishability factor is determined from the same.

The flowchart of the steps mentioned above is shown in Figure 3.5.
Figure 3.5 Flowchart for identifiability and distinguishability computations
3.7 FUNCTIONS CONSIDERED

The following ten random functions $F_1$ to $F_{10}$ have been considered and simulated for the reference structure and the five proposed structures using MATLAB coding for single-stuck-at, double-stuck-at, AND-bridging and OR-bridging faults and the results are given in the Chapters 4 to 9.

$F_1 = x_1 \oplus x_2x_3 \oplus x_1'x_2x_3$

$F_2 = x_1x_2 \oplus x_2'x_3 \oplus x_3'x_4 \oplus x_1x_2x_3$

$F_3 = x_1' \oplus x_2x_3'x_4 \oplus x_3'x_4' \oplus x_2'x_3 \oplus x_1x_4x_5$

$F_4 = x_1x_2' \oplus x_2x_3x_4' \oplus x_4x_5'x_6 \oplus x_2x_5 \oplus x_2'x_5' \oplus x_3'x_2x_1 \oplus x_4x_6$

$F_5 = x_1'x_2x_3 \oplus x_4x_5x_6 \oplus x_4'x_6'x_7 \oplus x_3x_5x_7$

$F_6 = x_1x_2'x_3 \oplus x_4'x_5x_6' \oplus x_7x_8' \oplus x_2x_6x_7' \oplus x_1'x_6 \oplus x_3'x_4 \oplus x_1x_5$

$\oplus x_4x_5' \oplus x_5x_7 \oplus x_8x_3x_1 \oplus x_3x_5'x_8$

$F_7 = x_1x_2'x_3' \oplus x_4x_5'x_6 \oplus x_7'x_8x_9 \oplus x_1'x_4'x_9' \oplus x_2x_5' \oplus x_3x_5$

$F_8 = x_1'x_2x_3 \oplus x_4'x_5'x_6 \oplus x_7x_8'x_9' \oplus x_1 \oplus x_6'x_7 \oplus x_8x_10$

$F_9 = x_1 \oplus x_2'x_3x_4' \oplus x_5'x_6x_7' \oplus x_8x_9x_10 \oplus x_10'x_11 \oplus x_1x_3x_9$

$F_{10} = x_1'x_2 \oplus x_3x_4'x_5 \oplus x_6x_7'x_8x_9 \oplus x_{10}x_{11} \oplus x_{12} \oplus x_1x_2x_3 \oplus x_4'x_7$

3.8 SUMMARY

The generalized network structure and the test set used for the detection of all single stuck-at, double stuck-at and bridging faults have been explained. The logic used for control function assignment and its ability in distinguishing distinguishability factors and identifiability factors have been detailed. The network structure and test vectors as discussed above serve as the reference structure for the detection of the four types of faults and are illustrated in Chapter 4.