A BIST Approach to Delay Fault Testing with Reduced Test Length

Bernd Wurth
Institute of Electronic Design Automation
Technical University of Munich
80290 Munich, Germany

Karl Fuchs
Siemens Mobile Radio Networks
81359 Munich, Germany

Abstract – *A cost-effective built-in self testing (BIST) method for the detection of delay faults is presented. A multiple-input signature register (MISR) with a constant parallel input vector is used as a test pattern generator.*

To reduce the test length of the MISR, a two-step approach is proposed. First, deterministic delay test generation is employed to determine a set of two-pattern tests which detect all testable path delay faults. Second, a minimal number of constant MISR input vectors is calculated such that the state sequences generated by the MISR include the pre-determined test set. The second step is formulated as a set covering problem.

As the number of MISR input vectors may be exponential in the number of stages of the MISR, their calculation and the set covering are performed implicitly with BDDs. Experimental results reveal that in almost all considered cases a maximum robust path delay fault coverage is obtained with less than 100 MISR input vectors.

1 Introduction

Aggressive statistical timing and synthesis tools are used to optimize the clock rate of high performance circuits. It was shown [1] that many paths in optimized circuits have a propagation delay close to the maximum circuit delay. To assure a high quality level of those circuits, delay testing becomes mandatory. Detecting a delay fault requires two distinct input vectors $\langle V, W\rangle$. First, the initialization vector V is applied to the circuit. After all signals have assumed their initial value, the propagation vector W is applied and the circuit outputs are sampled at clock speed.

Two main problems are associated with delay fault models leading to high costs of testing. First, long test sequences may be required to detect all testable delay faults. In large circuits the latter may hold even if test set reduction techniques are utilized. Second, expensive testers are required to apply the delay tests at normal clock speed to circuits.

An alternative technique is BIST which provides a low-cost test solution by building the tester inside the chip. To exploit BIST for delay fault testing, several BIST schemes targeting the generation of two-pattern tests have been proposed in the literature. These schemes differ with respect to the achieved fault coverage, the area overhead and the test length.

Exhaustive BIST test pattern generators (TPGs) were proposed [2,3,4,5]. Since the exhaustive two-pattern count for an n-input circuit under test (CUT) is $2^n \cdot (2^n - 1)$, exhaustive TPG methods are limited to CUTs with a small number of inputs.

A pseudo-exhaustive transition test using a $2n$-stage linear feedback shift register (LFSR) for an n-input CUT was proposed [3]. The test length of this method is $2^{2n} - 1$. To reduce the test length to $n \cdot 2^n$, pseudo-exhaustive adjacency testing (PEAT) was introduced [6]. PEAT gives a maximum robust path delay fault coverage [7], but due to an n-stage non linear feedback shift register (NFSR) and an additional n-stage register its area overhead is high. Recently [5] it was shown that a multiple input signature register (MISR) with only n stages allows an exhaustive two-pattern test in pipelined circuits.

To shorten the test time, m-stage LFSRs that provide a maximum test pattern-pair coverage with TPG size $m < 2n$ were identified [4]. However, the achievable robust delay fault coverage can only be determined via fault simulation in a post processing step. If the fault coverage is poor, further selection steps with increased TPG size may be required. A quite different approach to test set reduction is based on a pre-determined set of two-pattern tests detecting all testable delay faults [3,5]. In [3], the feedback function of an NFSR is designed to generate a given set of test pattern pairs. The time consuming determination of the NFSR, however, limits its applicability. In [5] it is proposed to calculate a minimum number of MISR input vectors such that the state sequences generated by the MISR include the pre-determined test set. The MISR input vectors are computed by clique covering.

In this paper a sophisticated test set reduction technique based on [5] is proposed. It starts from a given set of two-pattern tests. In particular, it will be shown that the determination of a minimal number of MISR inputs is a set covering problem rather than a clique covering problem. The set covering problem is efficiently solved using Binary Decision Diagrams (BDDs) [8] in all steps. Experimental results reveal that the test length can be reduced drastically compared with exhaustive testing. Our technique combines complete delay fault coverage with a small test length and low area overhead.

The paper is organized as follows. Section 2 contains some preliminaries on Boolean functions, BDDs and MISRs. The problem formulation and our solution strategy is given in Section 3. Our BDD-based approach is presented in detail in Section 4 and 5. In Section 6, experimental results are discussed. We give conclusion and directions for future work in Section 7.

2 Preliminaries

2.1 Boolean Functions

We are dealing with Boolean functions $f : B^n \to B$, where $B = \{0, 1\}$. Let $\mathbf{y} = (y_0, \ldots, y_{n-1})$ denote a vector of Boolean variables y_i. The *existential quantification* $\exists_{y_i}$ of a Boolean function $f(\mathbf{y})$ with respect to the variable y_i is defined by $\exists_{y_i} f(\mathbf{y}) = f(y_0, \ldots, y_{i-1}, 0, \ldots) + f(y_0, \ldots, y_{i-1}, 1, \ldots)$. The existential quantification of a Boolean function $f(\mathbf{y}, \mathbf{d})$ with respect to the variable vector $\mathbf{y} = (y_0, \ldots, y_{n-1})$ is given by $\exists_{\mathbf{y}} f(\mathbf{y}, \mathbf{d}) = \exists_{y_0} \exists_{y_1} \ldots \exists_{y_{n-1}} f(\mathbf{y}, \mathbf{d})$. The *density* $\rho(f)$ of a Boolean function $f : B^n \to B$ is the number of minterms in the onset of f divided by 2^n.

2.2 MISR with Constant Parallel Input

The next state **y'** and current state **y** of an n-stage MISR (see Fig. 1) with a constant input vector $\mathbf{d} = (d_0, ..., d_{n-1})$ can be related by the transfer function

$$\begin{pmatrix} y'_0 \\ y'_1 \\ \cdot \\ \cdot \\ \cdot \\ y'_{n-1} \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^{n-1} h_i \cdot y_{n-1-i} \oplus d_0 \\ y_0 \oplus d_1 \\ \cdot \\ \cdot \\ \cdot \\ y_{n-2} \oplus d_{n-1} \end{pmatrix} \qquad (1)$$

where $h_0 = 1$, $h_i \in \{0, 1\}$ for $1 \leq i \leq n-1$, and $\sum_{i=0}^{n-1} x_i$ being the addition modulo 2. The characteristic polynomial of the MISR with input vector $\mathbf{d} = (0, ..., 0)$ is given by $f_0(x) = 1 + \sum_{i=0}^{n-1} h_i \cdot x^{n-i}$. In [5] it has been shown that for each input vector **d** there exists a *maximum period* of length $2^n - 1$ if $f_0(x)$ is primitive [9] . The sequence of $2^n - 1$ states generated by the MISR with input vector **d** is called *maximum length sequence* (MLS) S_d.

Another important property was also proven in [5]:

Theorem 1 *Let $f_0(x)$ be primitive and let S_d and $S_{d'}$ denote the MLSs for two distinct inputs* **d** *and* **d'** *of the MISR. Then, two consecutive states $\langle \mathbf{y}, \mathbf{y'} \rangle$ that are contained in S_d will not be repeated in $S_{d'}$.*

According to Theorem 1, an exhaustive two-pattern test can be generated with an n-stage MISR if all 2^n input vectors are used and every single input vector **d** is held constant while the MISR cycles through its MLS S_d. Let us, e.g., consider a 3-stage MISR with transfer function

$$\begin{pmatrix} y'_0 \\ y'_1 \\ y'_2 \end{pmatrix} = \begin{pmatrix} y_2 \oplus y_0 \oplus d_0 \\ y_0 \oplus d_1 \\ y_1 \oplus d_2 \end{pmatrix} \qquad (2)$$

and the primitive polynomial $f_0(x) = x^3 \oplus x \oplus 1$. Fig. 2 depicts the eight MLSs for the eight input vectors **d**. The first variable y_0 represents the most significant bit, i.e., $\mathbf{y} = (100)$ is state 4. As an example, consider

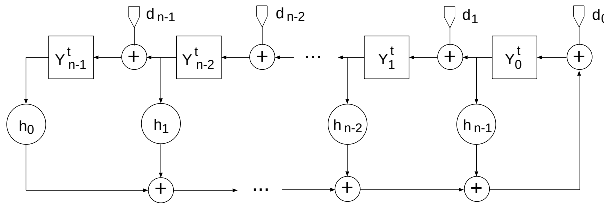

Figure 1: MISR with constant input vector $\mathbf{d} \in B^n$.

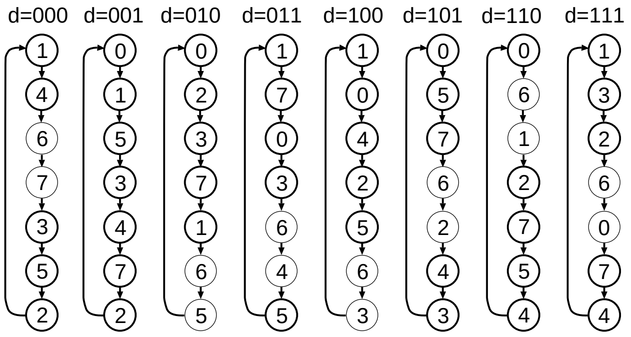

Figure 2: All MLSs representing an exhaustive two-pattern test.

the current state 6. All possible next states 0-7 except 6 are generated by the MISR if all input combinations except $\mathbf{d} = (001)$ are used. The same holds for all other current states.

2.3 Test Application

Since the test length of an exhaustive two-pattern test, as shown before, is $2^n(2^n - 1)$, a test set reduction method is required. In general, only a fraction of all constant input vectors are needed for complete fault coverage. Our algorithm described in the following sections determines a minimal number of inputs **d** such that the generated S_d's detect all testable delay faults in a given CUT. Fig. 3 shows the test application. For each input vector **d**, the MISR generates $2^n - 1$ test pattern pairs applied to the CUT.

If there is no direct access to the MISR inputs (e.g. in a piplined data path), each calculated input vector **d** has to be justified through the previous circuit C (see Fig. 3). The resulting vector **i** is serially loaded into the previous MISR and held constant for $2^n - 1$ clock cycles. To accomplish this, either two clocks with different periods or an additional mode is required to freeze the content of the previous MISR. As will be described in Subsection 5.3, our method is capable of calculating only those input vectors **d** that are valid outputs of C.

The constant input vectors **d** (or **i**, if the d-vectors are justified through the previous circuit C) can be stored in a simple read-only memory.

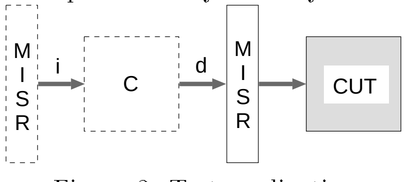

Figure 3: Test application.

3 Problem Formulation and Solution Approach

Let $\mathcal{M}_{test} = \{tp_1, \ldots, tp_m\}$ with $tp_j = \langle V_j, W_j \rangle$ be a set of deterministically generated two-pattern tests which detect all testable delay faults in the CUT. Then the problem is:

Determine a minimum cardinality set $\mathcal{M}_{MLS}$ of maximal length sequences S_d such that each test pattern pair $tp_j \in \mathcal{M}_{test}$ is contained in at least one $S_d \in \mathcal{M}_{MLS}$.

In general, V_j and W_j of tp_j are incompletely specified. Hence, a large number of MLSs may contain tp_j after proper assignment of don't care values. Since a fully defined vector pair is contained in exactly one MLS (see Theorem 1), an incompletely specified tp_j is included in 2^{p+q} MLSs, where p and q is the number of don't cares in V_j and W_j, respectively.

Definition 1 *A MLS that contains tp_j with properly specified don't cares is called* permissible *for the two-pattern test tp_j.*

The problem can be solved in two stages. First, the set $\mathcal{P}^j_{MLS}$ of all permissible MLSs is calculated for each two-pattern test $tp_j \in \mathcal{M}_{test}$. Note that $1 \leq |\mathcal{P}^j_{MLS}| \leq 2^n$. Second, a set covering problem has to be solved. Given the sets of permissible MLSs, determine a minimum cardinality set $\mathcal{M}_{MLS} = \{S_{d_1}, \ldots, S_{d_r}\} \subseteq \bigcup_{tp_j \in \mathcal{M}_{test}} \mathcal{P}^j_{MLS}$ such that for each $tp_j \in \mathcal{M}_{test}$ at least one of the MLSs $S_{d_1}, \ldots, S_{d_r}$ is permissible. If the inputs $\mathbf{d}_1, \ldots, \mathbf{d}_r$ are applied to the MISR, the MLSs $S_{d_1}, \ldots, S_{d_r}$ are generated and complete delay fault coverage is achieved in the CUT.

Both stages have not been thoroughly addressed in [5]. First, no systematic method is given to compute permissible MLSs. Second, the covering problem is formulated as a minimum clique covering problem. There is a node for each two-pattern test, and an edge between two nodes if the corresponding two-pattern tests are contained in one MLS. However, the two-pattern tests of a clique are not necessarily contained in a single MLS. Therefore, each clique found by minimum clique covering actually must be checked if its two-pattern tests are contained in a single MLS, and if not the clique must be postprocessed. Furthermore, even medium-size circuits have more than a thousand two-pattern tests and millions of possible MLSs such that explicit covering techniques would hardly succeed.

Figure 4 sketches the effect of both stages of our algorithm. The horizontal dimension displays the length of a single MLSs, the vertical dimension the number of MLSs needed for complete robust delay fault coverage. The whole rectangle represents the test length of an exhaustive two-pattern test as mentioned in Subsection 2.2 and shown in Fig. 2.

Solving the first stage, i.e., calculation of permissible MLSs for each two-pattern test, makes it possible to choose an arbitrary permissible MLS for each two-pattern test. This would reduce the test length from

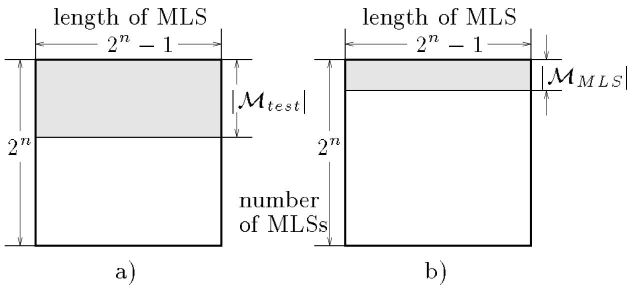

Figure 4: Test length reduction by a) calculation of permissible MLSs b) subsequent set covering.

$2^n \cdot (2^n - 1)$ to $|\mathcal{M}_{test}| \cdot (2^n - 1)$, as shown by the shaded area in Figure 4 a). Note that $|\mathcal{M}_{test}|$ typically is much smaller than 2^n. E.g., for the industrial circuit *ind1* we have $|\mathcal{M}_{test}| = 1685$, whereas 2^n is larger than $6.7 \cdot 10^7$.

The solution of the second stage, i.e., the set covering problem, further reduces the test length from $|\mathcal{M}_{test}| \cdot (2^n - 1)$ to $|\mathcal{M}_{MLS}| \cdot (2^n - 1)$ as shown in Figure 4 b). Again for circuit *ind1*, we have a reduction from $|\mathcal{M}_{test}| = 1685$ to $|\mathcal{M}_{MLS}| = 62$.

4 Generation of All Permissible MLSs

In this section we show how to implicitly compute the set of all permissible MLSs for a two-pattern test, using the characteristic function of the MISR.

As shown in Equ. (1), the function of the MISR output y'_0 is

$$y'_0 = \sum_{i=0}^{n-1} h_i \cdot y_{n-1-i} \oplus d_0 \,. \tag{3}$$

We define an individual *characteristic function* $\Phi_0(y'_0, \mathbf{y}, d_0)$ which is equal to 1 iff the input and output values of the MISR output y'_0 are consistent,

$$\Phi_0(y'_0, \mathbf{y}, d_0) = y'_0 \overline{\oplus} \left(\sum_{i=0}^{n-1} h_i \cdot y_{n-1-i} \oplus d_0 \right) , \tag{4}$$

where $\overline{\oplus}$ denotes XNOR.

The function of any other MISR output $y'_k, k = 1, \ldots, n-1$, was given in Equ. (1) as

$$y'_k = y_{k-1} \oplus d_k \,, \tag{5}$$

thus its individual characteristic function $\Phi_k(y'_k, \mathbf{y}, d_k)$ is

$$\Phi_k(y'_k, \mathbf{y}, d_k) = y'_k \overline{\oplus} (y_{k-1} \oplus d_k) \,. \tag{6}$$

Combining the individual characteristic functions of each output, we obtain the characteristic function Φ_{MISR} of the MISR, which yields 1 for any consistent assignment of input and output values of the MISR:

$$\Phi_{MISR}(\mathbf{y'}, \mathbf{y}, \mathbf{d}) = \prod_{k=0}^{n-1} \Phi_k(y'_k, \mathbf{y}, d_k) \tag{7}$$

Now let us consider a specific test pattern pair $tp_j \in \mathcal{M}_{test}$, and set $\mathbf{y} = V_j$ and $\mathbf{y'} = W_j$. In general, not all the variables $y'_0, \ldots, y'_{n-1}$ and $y_0, \ldots, y_{n-1}$ are specified. We ask for values of these variables so that the MISR input and output value assignments are consistent. If there exists a combination of values for $y'_0, \ldots, y'_{n-1}$ and $y_0, \ldots, y_{n-1}$ so that $\Phi_{MISR}(\mathbf{y'}, \mathbf{y}, \mathbf{d}) = 1$ then we have found a consistent value assignment. Performing existential quantification with respect to the vectors **y'** and **y** on the characteristic function Φ_{MISR} of the MISR, we obtain a function denoted $pf_j(\mathbf{d})$,

$$pf_j(\mathbf{d}) = \exists_{\mathbf{y'}, \mathbf{y}} \; \Phi_{MISR}(\mathbf{y'} = W_j, \mathbf{y} = V_j, \mathbf{d}), \qquad (8)$$

which depends on the d-variables only. The onset of $pf_j(\mathbf{d})$ consists of all those d-vectors for which a consistent value assignment to the don't cares of the test pattern pair tp_j exists. Therefore, the onset of $pf_j(\mathbf{d})$ uniquely defines the set $\mathcal{P}^j_{MLS}$ of permissible MLSs for the test tp_j: $\mathcal{P}^j_{MLS} = \{S_{d_i} | pf_j(\mathbf{d}_i) = 1\}$. The function of permissibles $pf_j(\mathbf{d})$ is called the *p-function* for test tp_j.

Example:
For the 3-stage MISR with transfer function (2), see Subsection 2.2, we get $\Phi_{MISR}(\mathbf{y'}, \mathbf{y}, \mathbf{d}) = (y'_0 \overline{\oplus} (y_2 \oplus y_0 \oplus d_0)) \cdot (y'_1 \overline{\oplus} (y_0 \oplus d_1)) \cdot (y'_2 \overline{\oplus} (y_1 \oplus d_2))$.
For $tp_j = \langle X01, 011 \rangle$ we have $y'_0 = 0, y'_1 = 1, y'_2 = 1, y_0 = X, y_1 = 0, y_2 = 1$ and therefore

$$pf_j(\mathbf{d}) = \exists_{\mathbf{y}, \mathbf{y'}} \; \Phi_{MISR}(\mathbf{y'}, \mathbf{y}, \mathbf{d})$$
$$= \exists_{y_0} \; ((0 \overline{\oplus} (1 \oplus y_0 \oplus d_0))(1 \overline{\oplus} (y_0 \oplus d_1))(1 \overline{\oplus} (0 \oplus d_2)))$$
$$= \exists_{y_0} \; ((y_0 \oplus d_0) \cdot (y_0 \oplus d_1) \cdot (d_2))$$
$$= d_0 d_1 d_2 + \overline{d_0}\,\overline{d_1} d_2.$$

There are two permissible MLSs for tp_j: S_{111} and S_{001}, thus $\mathcal{P}^j_{MLS} = \{S_{001}, S_{111}\}$.

5 Implicit Set Covering

Each d-vector which is in the onset of a p-function $pf_j(\mathbf{d})$ represents a MLS permissible for tp_j. Our goal is to determine a minimum cardinality set of d-vectors such that for each $tp_j \in \mathcal{M}_{test}$ at least one of these d-vectors is in the onset of $pf_j(\mathbf{d})$.

Obviously, we have to solve a set covering problem. In the table associated with this set covering problem, there is a row for each test pattern pair tp_j and a column for each d-vector. The set covering problem is to find a minimum cardinality subset of columns that cover all the rows [10].

We know of two BDD-based algorithms for a set covering problem where the rows of the table are given by functions, as it is the case with the p-functions in our problem. The first [11] tackles the more general binate covering problem. However, each column of the covering table is represented by a variable, which is not applicable in our case where the number of columns is 2^n. An exact set covering algorithm was proposed in [12] in a different context. In our case, the entries of the table depend on the particular two-pattern tests which are determined by automatic test pattern generation (ATPG). Since ATPG methods usually do not provide an optimal test set with respect to size and a maximum number of don't cares, the optimality of an exact set covering step has only restricted practical value.

Table 1: Example for row dominance

tp_j	$\langle V_j, W_j \rangle$	$pf_j(\mathbf{d})$	row dominance
tp_1	(00X, 10X)	11001100	$tp_1 \preceq tp_4$
tp_2	(10X, 00X)	00110011	$tp_2 \preceq tp_3$
tp_3	(01X, 111)	00100010	
tp_4	(111, 011)	10000000	
tp_5	(01X, 11X)	00110011	$tp_5 \preceq tp_3$
tp_6	(X01, 011)	01000001	
tp_7	(01X, 001)	10001000	$tp_7 \preceq tp_4$
tp_8	(00X, 01X)	00110011	$tp_8 \preceq tp_3$
tp_9	(01X, 00X)	11001100	$tp_9 \preceq tp_4$
tp_{10}	(10X, 11X)	11001100	$tp_{10} \preceq tp_4$
tp_{11}	(X00, X01)	01010101	$tp_{11} \preceq tp_6$
tp_{12}	(X01, X00)	10101010	$tp_{12} \preceq tp_3$

Table 2: After exploiting row dominance

tp_j	$pf_j(\mathbf{d})$
tp_3	00100010
tp_4	10000000
tp_6	01000001

Therefore we use a computationally inexpensive heuristic algorithm to find a minimal column covering. Our set covering algorithm is characterized by three steps. First, we exploit row dominance. We then determine essential columns [10], which is not described here due to space limitations. Finally, the resulting table is covered greedily by columns which cover a maximum number of rows. All steps are performed with BDDs.

5.1 Exploiting Row Dominance

The test pattern pair tp_j associated with row j is said to dominate test pattern pair tp_k associated with row k, if each MLS permissible for tp_j is also permissible for tp_k. The dominance relation between test pattern pairs can be expressed using the p-functions:

$$tp_k \preceq tp_j \iff (\overline{pf_j(\mathbf{d})} + pf_k(\mathbf{d}) = 1) \qquad (9)$$

Dominated test pattern pairs are deleted because any d-vector which is in the onset of the p-function $pf_j(\mathbf{d})$ of tp_j will also be in the onset of the p-function $pf_k(\mathbf{d})$ of tp_k. Deleting dominated test pattern pairs reduces the problem size without sacrificing optimality.

An example is shown in Table 1. Twelve test pattern pairs $tp_j = \langle V_j, W_j \rangle$ have been generated deterministically for the circuit shown in Fig. 5 of [5]. The respective V_j and W_j patterns are given in column 2. The truth table of each $pf_j(\mathbf{d})$ is shown in column 3 as a row vector, where the leftmost value is $pf_j(000)$. For example, $pf_1(000) = 1$ and $pf_1(111) = 0$. Note that tp_6 is the example given at the end of Section 4.

The complete column 3 represents the table of the set covering problem. The row dominance relations are given in column 4. Nine of the twelve test pattern pairs are dominated so that only three pattern pairs need to be considered in the following step of the set covering. The resulting table after exploiting row dominance is shown in Tab. 2. Three columns are necessary to cover the remaining three rows.

5.2 Selecting a Maximum Column

To determine a column which covers a maximum number of rows, the covering matrix is constructed as a BDD. Let there be m dominating p-functions after exploiting row dominance. We assign a unique code $\mathbf{c} \in B^z, z \geq \lceil ld\ m \rceil$ to each of these p-functions. Then, the function CM representing the covering matrix is

$$CM(\mathbf{c}, \mathbf{d}) = \sum_{j=1}^{m} \mathbf{c}_j \cdot pf_j(\mathbf{d}). \tag{10}$$

The maximum covering column is determined as suggested in [13]. The variables in $\mathbf{d}$ must be ordered before the variables in $\mathbf{c}$. Let the variables in $\mathbf{d}$ have index $0, \dots, n-1$, and let the variables in $\mathbf{c}$ have index $\geq n$. By *top_c_nodes* we denote the set of BDD nodes which have an index $\geq n$ and at least one predecessor with index $< n$.

For $\nu_j \in$ *top_c_nodes*, let $h_j(\mathbf{d})$ be the function having as its onset all those $\mathbf{d}$-product terms which correspond to paths from the root node of the BDD to ν_j, and let $f_j(\mathbf{c}) = BDD(\nu_j)$ be the Boolean function represented by node ν_j. Then, for each ν_j the function $h_j(\mathbf{d})$ represents all the columns which cover the same rows. These rows are represented by function $f_j(\mathbf{c})$.

The density ρ is calculated for each BDD node $\nu_j \in$ *top_c_nodes*. We choose $\nu_j \in$ *top_c_nodes* such that the density of its function, i.e. $\rho(BDD(\nu_j))$, is maximum. Since all $\mathbf{d}$-minterms in the onset of $h_j(\mathbf{d})$ cover the same rows, we arbitrarily select one of these minterms. This minterm $\mathbf{d}$ defines the MLS S_d, which is assigned to the solution set $\mathcal{M}_{MLS}$.

The rows covered by the selected column are removed, i.e., the function CM is updated, and the next maximum covering column is calculated.

5.3 Justifying Constant Input Vectors

If each d-vector must be justified through the previous circuit C as described in Subsection 2.3, we first compute the range $R_C(\mathbf{d})$ of the multiple-output function implemented by circuit C. Range computation can be done as suggested in [14].

Before we exploit row dominance, each p-function $pf_j(\mathbf{d})$ is simply multiplied with the range $R_C(\mathbf{d})$: $pf_j'(\mathbf{d}) = pf_j(\mathbf{d}) \cdot R_C(\mathbf{d})$. Thus we do not further consider all those MLS S_d for which the d-vector cannot be produced by the preceding circuit C.

6 Experimental Results

To evaluate our method, we used ISCAS-85 and ISCAS-89 benchmark circuits as well as some industrial examples. Only the combinational parts of the circuits were used. The second column of Table 3 shows the number of inputs $\sharp$**Inputs** $= n$ of the circuits. Since the test length for each calculated input vector $\mathbf{d}$ is $2^n - 1$, test application time can become too large for circuits with $n > 30$ inputs. Therefore, only circuits with $n < 30$ have been selected.

In a first step, we determined robust two-pattern tests for all path delay faults using DYNAMITE [15]. The number of generated two-pattern tests, $\sharp$**TPs** $= |\mathcal{M}_{test}|$, is given in column 3. Note that the random assignment of values 0 or 1 to the don't cares after ATPG was omitted to make many MLSs permissible for a two-pattern test.

In a second step, we computed the permissible MLSs as described in Section 4 for each two-pattern test. Then, row dominance according to Equ. (9) was exploited. The resulting number of dominating two-pattern tests, $\sharp$**Rows**, is given in column 4 of Table 3. Solution of the set covering problem, as discussed in Section 5, yields a minimal number of maximum length sequences $\sharp$**MLS** $= |\mathcal{M}_{MLS}|$ which is given in column 5. Except for circuit *ind2*, the final number of MLSs is always below 100. It can be seen from the result table that most of the reduction from $|\mathcal{M}_{test}|$ to $|\mathcal{M}_{MLS}|$ is due to the initial exploitation of row dominance.

Column 6 displays the quotient $2^n/\sharp$**TPs**, which is the test length reduction achieved by the first stage of our BDD-based algorithm as shown in Fig. 4a). This value ranges from below 3 for *s27* up to over 10^5 for *ind3*. Apparently, the reduction is more drastic for circuits with a large number of inputs.

Column 7 shows the average number of two-pattern tests per maximum length sequence, $\sharp$**TPs**/$\sharp$**MLSs**, which is the test length reduction obtained by the second stage of our algorithm as shown in Fig. 4b). The reduction ranges from 6.3 for *ind2* up to 46.5 for *ind3*. We can conclude that both stages of our BDD-based algorithm significantly contribute to a reduction of the test length. The last two columns show the CPU time in seconds on a DecStation 5000/200 (22 MIPS) spent for ATPG and our BDD-based calculation of a minimal set of MLSs.

A comparison of the number of MLSs for *s386* and *s444* illustrates that a smaller number of two-pattern tests need not cause a smaller number of MLSs. The final number of MLSs also depends on how many permissible MLSs exist for each two-pattern test: experimental data reveals that the average density of the dominating p-functions of *s386* is 0.024 while the average density of the dominating p-functions of *s444* is 0.093, i.e., almost four times as large. Maximizing the number of don't cares in the test patterns can therefore be viewed as an additional objective during ATPG.

7 Conclusion

We have developed a new BIST method for the detection of delay faults. The method is based on a MISR generating a maximum length sequence of patterns for each constant input vector.

The test length is determined by the number of patterns of a maximum length sequence and the number of constant input vectors. In this work, we concentrated on minimizing the number of constant input vectors. Using deterministically generated two-pattern tests, we calculated the set of permissible maximum length sequences for each two-pattern test. The calculation is performed implicitly with BDDs.

circuit	♯Inputs	♯TPs	♯Rows	♯MLSs	2^n/♯TPs	♯TPs/♯MLS	CPU/sec	
							ATPG	*MLS*
s27	7	50	10	6	2.6	8.3	3.7	0.7
s208	18	284	51	26	923.0	10.9	5.4	14.0
s298	17	343	35	17	382.1	20.2	5.5	7.2
s386	13	413	85	49	19.8	8.4	7.5	17.8
s444	24	586	109	16	$2.8 \cdot 10^4$	36.6	8.7	361.4
s510	25	729	51	34	$4.6 \cdot 10^4$	21.4	13.5	16.9
s526	24	694	71	37	$2.4 \cdot 10^4$	18.8	8.3	71.8
s820	23	980	140	71	$8.6 \cdot 10^3$	13.8	19.5	88.7
s832	23	984	148	71	$8.5 \cdot 10^3$	13.9	20.2	104.0
s1488	14	1875	198	76	8.7	24.7	82.3	112.9
s1494	14	1882	187	73	8.7	25.8	84.0	109.5
ind1	26	1685	165	62	$4.0 \cdot 10^4$	27.2	33.9	166.6
ind2	24	1155	246	183	$1.5 \cdot 10^4$	6.3	26.0	899.4
ind3	26	186	7	4	$3.6 \cdot 10^5$	46.5	4.4	4.5
ind4	25	1393	113	66	$2.4 \cdot 10^4$	21.1	556.8	680.2

Table 3: Calculating the minimal cardinality set of MLSs

We obtain sets of permissible maximum length sequences and use a BDD-based set covering algorithm to compute a minimal number of maximum length sequences, and thus a minimal number of constant input vectors. If each of these input vectors is held constant while the MISR cycles through its maximum length sequence, complete fault coverage is obtained in the circuit under test.

Our approach can be summarized as follows. First, because a n-stage MISR is used in our BIST approach, area overhead is low compared with other approaches which require a $2n$-stage test pattern generator. Second, the use of deterministically generated two-pattern tests allows complete delay fault coverage with moderate test length. Finally, to handle the huge number of permissible maximum length sequences, the computation is based on efficient implicit BDD-techniques.

So far we only considered the vertical dimension of Fig. 4 to reduce the test length. As each maximum length sequence consists of $2^n - 1$ patterns, our approach is still confined to circuits with less than 30 inputs. To be able to also handle large circuits without partitioning, our future work will focus on the horizontal dimension of Fig. 4. This means that we will try to determine a set of sequences of length $< 2^n - 1$ such that the test length, as given by the shaded area in Fig. 4, is minimized.

Acknowledgment

The authors would like to thank Alfred Kölbl for doing a part of the implementation. The first author was supported by a Ernst von Siemens grant given by Siemens AG.

References

[1] T. W. Williams, B. Underwood, and M. R. Mercer, "The Interdependence Between Delay-Optimization of Synthesized Networks and Testing," *28th ACM/IEEE Design Automation Conference DAC*, pp. 87–92, 1991.

[2] K. Furuya and E. J. McCluskey, "Two-Pattern Test Capabilities of Autonomous TPG Circuits," *IEEE International Test Conference*, pp. 704–711, 1991.

[3] C. W. Starke, "Build-In Test for for CMOS Circuits," *IEEE International Test Conference*, pp. 309–314, 1984.

[4] C. A. Chen and S. K. Gupta, "BIST Test Pattern Generators for Stuck-Open and Delay Testing," *European Design Automation Conference EDAC 1994*, pp. 289–296, 1994.

[5] A. Vuksic and K. Fuchs, "A New BIST Approach for Delay Fault Testing," *European Design Automation Conference EDAC 1994*, pp. 284–288, 1994.

[6] G. L. Craig and C. R. Kime, "Pseudo-Exhaustive Adjacency Testing: A BIST Approach for Stuck-Open Faults," *IEEE International Test Conference*, pp. 126–137, 1985.

[7] G. L. Smith, "Model for Delay Faults Based upon Paths," *IEEE International Test Conference*, pp. 342–349, 1985.

[8] R. E. Bryant, "Graph-Based Algorithms for Boolean Function Manipulation," *IEEE Transactions on Computers*, vol. 35, no. 8, pp. 677–691, 1986.

[9] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications.* Englewood Cliffs, N.J.: Prentice Hall, 1983.

[10] R. K. Brayton, G. D. Hachtel, C. T. McMullen, and A. L. Sangiovanni-Vincentelli, *Logic Minimization Algorithms for VLSI Synthesis.* Boston, MA: Kluwer Academic Publishers, 1984.

[11] S.-W. Jeong and F. Somenzi, "A New Algorithm for the Binate Covering Problem and its Application to the Minimization of Boolean Relations," *IEEE/ACM International Conference on Computer-Aided Design ICCAD*, pp. 417–420, 1992.

[12] Y. Matsunaga, "MINT-An Exact Algorithm for Finding Minimum Test Set," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E76-A, no. 10, pp. 1652–1658, 1993.

[13] T. Kam, T. Villa, R. Brayton, and A. Sangiovanni-Vincentelli, "A Fully Implicit Algorithm for Exact State Minimization," *31th ACM/IEEE Design Automation Conference DAC*, 1994.

[14] O. Coudert and J. C. Madre, "A Unified Framework for the Formal Verification of Sequential Circuits," *IEEE/ACM International Conference on Computer-Aided Design ICCAD*, pp. 126–129, 1990.

[15] K. Fuchs, F. Fink, and M. H. Schulz, "DYNAMITE: An Efficient Automatic Test Pattern Generation System for Path Delay Faults," *IEEE Transactions on Computer-Aided Design*, vol. 10, no. 10, pp. 1323 – 1335, 1991.