Exhaustive Two-pattern Generation with Cellular Automata*

Sukumar nandi P. Pal Chaudhuri
Dept. of Comp. Sc. & Engg.
Indian Institute of Technology
Kharagpur - 721 302
INDIA

Samir Roy
Dept. of Comp. Sc.
NERIST
Nirjuli - 791 109
INDIA

M. Sharma
Inter Software & Technologies (India) Pvt. Ltd., NEPZ
NOIDA - 201 305
INDIA

Abstract

Two-patterns are required to test a transistor stuck-open fault or a delay fault within a combinational circuit. Cellular Automata (CA) has been proposed as a two pattern generator in the present work. For a 2n-cell CA structure, the condition to generate all possible exhaustive two-pattern n-bits have been investigated. Criteria to select the most desirable CA structure have also been laid down.

Key words : Cellular Automata, Built-In-Self-Test, Design-For-Testability, two-pattern test, stuck-open faults, delay faults.

1 Introduction

Built-In-Self-Test (BIST) and Design-For-Testability (DFT) have recently emerged as the state-of-art in VLSI design arena. The objective of BIST or DFT is to enhance testability of a VLSI circuit. Detection of transistor stuck-open faults and delay faults require application of two consecutive patterns rather than a single pattern. Following discussion elaborates the test generation problems.

I) S-open faults : Consider the 2-input NAND gate of Fig 1. Detection of each of the stuck-open fault indicated in the figure requires two consecutive patterns at the input one of which initializes the path through the fault concerned and the other is used to sensitize the fault in that path. The initializing pattern is such that the output is set to a value which is the complement of the value generated by the sensitizing pattern. Table 1 shows the two-patterns necessary for detecting the faults $f_1$ to $f_6$ marked in Fig 1.

II) Delay faults. Let an input pattern be applied on the circuit-under-test (CUT) at time $t_0$. If the delay of the CUT is $\delta$ then the effect of this input pattern should be manifested at time $t_0 + \delta$. A delay fault is said to have occurred if the effect does not propagate in due time so that the desired logic level is not available at the output at the predetermined time instant. Delay faults are caused by faulty or improper circuit components. It is obvious that detection of a delay fault also requires application of test patterns in two phases viz. initialization and sensitization. Consider the circuit shown in Fig 2. Detection of each

---

*This research work is partially supported by DOE sponsored project No. DE-NMC/SP-009

---

Figure 1: Different stuck-open faults in a 2-input NAND gate.

Table 1: Two-pattern for detecting faults in Fig 1.

<table>
<thead>
<tr>
<th>Faults</th>
<th>$f_1$</th>
<th>$f_2$</th>
<th>$f_3$, $f_4$</th>
<th>$f_5$, $f_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inputs</td>
<td>$x_0$</td>
<td>$x_1$</td>
<td>$x_0$, $x_1$</td>
<td>$x_0$, $x_1$</td>
</tr>
<tr>
<td>Initialization</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Sensitization</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

230
Figure 2: A circuit for delay fault test patterns.

of the delay faults indicated in the figure requires two consecutive patterns. Table 2 shows the two-patterns necessary for detecting the faults.

Table 2: Two-pattern for detecting faults in Fig 2

<table>
<thead>
<tr>
<th>Faults</th>
<th>( f_1 )</th>
<th>( f_2 )</th>
<th>( f_3 )</th>
<th>( f_4 )</th>
<th>( f_5 )</th>
<th>( f_6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initiation</td>
<td>010</td>
<td>110</td>
<td>001</td>
<td>011</td>
<td>010</td>
<td>011</td>
</tr>
<tr>
<td>Sensitization</td>
<td>110</td>
<td>010</td>
<td>011</td>
<td>001</td>
<td>110</td>
<td>100</td>
</tr>
</tbody>
</table>

Yarmolik[1] and Bardell[2] have investigated the application of pseudorandom sequences for testing VLSI circuits. Relationships between the characteristic polynomials and the cyclic structures of the generated sequences are investigated in [3] Furuya [4] has investigated the two-pattern test capabilities of arbitrary autonomous linear sequential circuits for use in BIST of CMOS circuits. They [4] introduced the concept of transition coverage and based on the transition matrix of the circuit, a method is shown to analyse of the state variable. Savir[5] has introduced the parameter called AC strength of a test generator and has proposed the input separation scheme that will allow an efficient AC test to be performed on the logic.

In this paper design of a linear circuit as an exhaustive two-pattern generator has been investigated analytically. A sufficient condition for exhaustive two-pattern generation has been derived. The linear circuit of additive CA has been shown to satisfy the sufficient condition. The simple, modular and cascadable structure of CA ideally suits for VLSI design.

Complete characterization of additive CA based on matrix algebraic tool has been reported in [7, 8]. The characteristic matrix \( T \) of a CA is written using the following formulation.

\[
T_{ij} = \begin{cases} 
1 & \text{if } j^{th} \text{ cell depends on the } i^{th} \text{ cell} \\
0 & \text{otherwise} 
\end{cases}
\]

Adding \( T \) to \( xI \) ( \( I \) is an \( n \times n \) identity matrix and \( x \) is an indeterminate) and evaluating the corresponding determinant gives an \( n \) degree polynomial, called the characteristic polynomial of the CA matrix \( T \). This is illustrated with the following example 1.

**Example 1:** A four cell hybrid CA with null boundary condition with the following rules \( < 90, 150, 90, 150 > \) applied from left to right, may be characterized by the following characteristic matrix

\[
T = \begin{bmatrix} 
0 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 
\end{bmatrix}
\]

\[
T + xI = \begin{bmatrix} 
x & 1 & 0 & 0 \\
1 & 1 + x & 1 & 0 \\
0 & 1 & x & 1 \\
0 & 0 & 1 & 1 + x 
\end{bmatrix}
\]

The characteristic polynomial is \( x^4 + x + 1 \).

Again, if the characteristic polynomial of a CA is primitive then that CA is called maximal length CA. Which can generate all states in a single cycle excluding all zero state.

2 Analysis of the exhaustive two-pattern generation capability of a CA.

For \( n \) bits there can be total of \( 2^n(2^n - 1) \) pairs of bits patterns. Since there are a maximum of \( 2^n \) bit patterns generated by a bit CA, \( 2^n(2^n - 1) \) number of two-patterns can be generated by a CA of \( 2n \) cells. As the CA cycles through its states the two-patterns are exhaustively generated only at \( n \) bit positions, out of the \( 2n \) bits. There are \( 2^{n}C_{n} \) ways of selecting those \( n \) bits, out of which only a few are capable of generating the two-patterns exhaustively. The combination of such bit positions depends on the interconnection of the CA i.e. the rule combination of the CA.

**Definition 1:** In a \( 2n \)-bit CA with \( n = 1, 2, 3, \ldots \), let \( v_{c0}, v_{c1}, \ldots, v_{cn-1} \) be \( n \) integers such that \( v_{ci} \leq 2n \) \( \forall i \), on which we are looking for an exhaustive two-pattern generation. The CA bit positions corresponding to these integers are called visible bits and the set \( V C = \{ v_{c0}, v_{c1}, \ldots, v_{cn-1} \} \) is the set of all visible bits. Similarly, \( OC = \{ o_{c0}, o_{c1}, \ldots, o_{cn-1} \} \) is set of obscure bits. \( VC \) and \( OC \) have the following properties:

i) \( VC \cap OC = \phi \)

ii) \( VC \cup OC = \{ 0, 1, \ldots, 2n - 1 \} \).

**Note:** For the sake of brevity, we shall implicitly assume that:

i) \( v_{ci} < v_{cj} \) if \( i < j \), and
ii) $o_{ij} < o_{ij}$ if $i < j$.

The corresponding rows of the characteristic matrix $T$ of the CA are termed to be visible rows and obscure rows, respectively.

If $S$ and $\bar{S}$ be the present and the next CA state then $S_v, S_o$ denote their respective visible parts. Similarly, $\bar{S}_v$ and $\bar{S}_o$ are the obscure parts. $S$ and $\bar{S}$ are related by the following equation

$$\bar{S} = T \times S \quad (1)$$

If we consider the visible part of the next state, we get,

$$\bar{S}_v = [T_{v_0}, \ldots, T_{v_{c-1}}, \ldots, T_{v_{c-1}}, \ldots] \times S \quad (2)$$

where $T_{v_i}$ is the $v_i$th row of the characteristic matrix $T$.

R.H.S of equation (2) can be split into two parts as follows:

$$\bar{S}_v = T_v \times S_v + T_o \times S_o \quad (3)$$

where,

$$T_v = \begin{bmatrix}
T_{v_0}(v_{c0}) & \cdots & T_{v_0}(v_{c_{c-1}}) \\
\vdots & \ddots & \vdots \\
T_{v_0}(v_{c_{c-1}})(v_{c0}) & \cdots & T_{v_0}(v_{c_{c-1}})(v_{c_{c-1}})
\end{bmatrix}_{n \times n}$$

and

$$T_o = \begin{bmatrix}
T_{v_0}(o_{c0}) & \cdots & T_{v_0}(o_{c_{c-1}}) \\
\vdots & \ddots & \vdots \\
T_{v_0}(o_{c_{c-1}})(o_{c0}) & \cdots & T_{v_0}(o_{c_{c-1}})(o_{c_{c-1}})
\end{bmatrix}_{n \times n}$$

Definition 2: Matrix $T_v$ and $T_o$ are named as the visibility matrix and the obscurity matrix respectively.

Example 2: Consider an 8-cell null boundary CA with rule $<90, 90, 90, 90, 90, 150, 150, 90>$. The next state function of this is represented by

$$\bar{S} = T \times S$$

Let $\{0, 2, 4, 6\}$ and $\{1, 3, 5, 7\}$ are the sets of all visible bits and obscure bits respectively. Then the next state function for visible bits can be written as

$$\begin{bmatrix}
\bar{s}_0 \\
\bar{s}_1 \\
\bar{s}_2 \\
\bar{s}_3 \\
\bar{s}_4 \\
\bar{s}_5 \\
\bar{s}_6 \\
\bar{s}_7
\end{bmatrix} =
\begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
s_0 \\
s_1 \\
s_2 \\
s_3 \\
s_4 \\
s_5 \\
s_6 \\
s_7
\end{bmatrix} =
\begin{bmatrix}
s_0 \\
s_1 \\
s_2 \\
s_3 \\
s_4 \\
s_5 \\
s_6 \\
s_7
\end{bmatrix} +
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
s_0 \\
s_1 \\
s_2 \\
s_3 \\
s_4 \\
s_5 \\
s_6 \\
s_7
\end{bmatrix}$$

or in general by

$$\bar{S}_v = T_v \times S_v + T_o \times S_o$$

Theorem 1: For a given $2n$-cell CA and a set of visible bits, an exhaustive two-pattern generation is ensured if the rank of the corresponding obscurity matrix is $n$.

Proof: An arbitrary two-pattern $S_v, \bar{S}_v$ is obtained if and only if the following equation is satisfied.

$$\bar{S}_v = T_v \times S_v + T_o \times S_o \Rightarrow T_v \times S_v = \bar{S}_v - T_o \times S_o = X \quad (4)$$

where $X$ is the $n$-dimensional column matrix $S_v - T_v \times S_v$. A solution for $S_v$ exists if

$$\text{rank}[T_v] = \text{rank}[T_v X]$$

and this is ensured if $\text{rank}[T_v] = n$.

Q.E.D.

Corollary 1: For any CA with characteristic matrix $T$:

$$T = \begin{bmatrix}
0 & 1 & 0 & \cdots & 0 & 0 \\
0 & 0 & 1 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 0 & 1 \\
0 & 0 & 0 & \cdots & 0 & 0
\end{bmatrix}_{2n \times 2n}$$

i.e.

$$T_{ij} = \begin{cases}
1 & \text{if } j = i + 1 \\
0 & \text{otherwise}
\end{cases}$$

the column $0, 2, 4, \ldots, 2(n-1)$ bits are exhaustive two-pattern generators.

Proof: The obscurity matrix is given by:

$$T_o = \begin{bmatrix}
1 & 0 & \cdots & 0 \\
0 & 1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 1
\end{bmatrix}_{n \times n}$$
Obviously, \( \text{rank}(T_o) = n \). Hence the corollary is proved by virtue of theorem 1.

Q.E.D.

The implication of the above corollary is that any left shifting shift register is capable of generating all \( n \)-bit two-patterns at bit positions 0, 2, 4,\( \cdots \) etc. The problem is that, in such a case we have to load the shift register with different seeds from time to time, which is costly. It would have been very convenient if we could load the generator just once and go on applying clock cycle such that all the two-patterns are eventually obtained. A maximal length CA is the most suitable structure for this purpose. Since it runs through all the states with single loading of a non-zero seed.

**Corollary 2:** A 2\( n \)-cell CA with any combination of Rule 90 and Rule 150 over the cells is capable of generating all two-patterns at the cells 0, 2, 4,\( \cdots \) etc.

**Proof:** The characteristic matrix of a 2\( n \)-cell CA with an arbitrary combination of Rule 90 and Rule 150 is given by:

\[
T = \begin{bmatrix}
    g_0 & 1 & 0 & \cdots & 0 \\
    1 & g_1 & 1 & \cdots & 0 \\
    0 & 1 & g_2 & \cdots & 0 \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    0 & 0 & 0 & \cdots & 1 & g_{2n-1}
\end{bmatrix}_{2n \times 2n}
\]

where

\[
g_i = \begin{cases} 
1 & \text{if the corresponding rule is 150} \\
0 & \text{otherwise}
\end{cases}
\]

Here

\[
T_o = \begin{bmatrix}
    T_{01} & T_{03} & \cdots \\
    T_{11} & T_{13} & \cdots \\
    \vdots & \vdots & \ddots \\
    T_{2(n-1)} & T_{2(n-1)} & \cdots
\end{bmatrix}
\]

\[
= \begin{bmatrix}
    1 & 0 & \cdots & 0 \\
    1 & 1 & \cdots & 0 \\
    \vdots & \vdots & \ddots & \vdots \\
    0 & 0 & \cdots & 1
\end{bmatrix}_{n \times n}
\]

Obviously, \( \text{rank}(T_o) = n \).

Q.E.D.

Among these, one with the least number of Rule 150 should be selected. Such a selection criterion leads to reduced circuit complexity & chip area because (i) routing area is reduced, and (ii) number of 3-input EX-OR is minimised. Further, use of maximum length CA ensures that the desired pattern set can be generated with loading of a seed.

Simulation result shows that exhaustive two-patterns are generated not only at the even numbered cell positions. Table 3 indicates the cell positions of the CA, for different lengths, where exhaustive two-patterns are being generated.

<table>
<thead>
<tr>
<th>( n )</th>
<th>( 4 )</th>
<th>( 6 )</th>
<th>( 8 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cell-positions</td>
<td>0.2, 0.24, 0.24, 0.24, 0.24</td>
<td>0.2, 0.24, 0.24, 0.24, 0.24</td>
<td>0.2, 0.24, 0.24, 0.24, 0.24</td>
</tr>
<tr>
<td>1, 2, 3, 4, 5</td>
<td>0.2, 0.24, 0.24, 0.24, 0.24</td>
<td>0.2, 0.24, 0.24, 0.24, 0.24</td>
<td>0.2, 0.24, 0.24, 0.24, 0.24</td>
</tr>
<tr>
<td>1, 2, 3, 4, 5, 6</td>
<td>0.2, 0.24, 0.24, 0.24, 0.24, 0.24</td>
<td>0.2, 0.24, 0.24, 0.24, 0.24, 0.24</td>
<td>0.2, 0.24, 0.24, 0.24, 0.24, 0.24</td>
</tr>
<tr>
<td>1, 2, 3, 4, 5, 6, 7</td>
<td>0.2, 0.24, 0.24, 0.24, 0.24, 0.24, 0.24</td>
<td>0.2, 0.24, 0.24, 0.24, 0.24, 0.24, 0.24</td>
<td>0.2, 0.24, 0.24, 0.24, 0.24, 0.24, 0.24</td>
</tr>
<tr>
<td>1, 2, 3, 4, 5, 6, 7, 8</td>
<td>0.2, 0.24, 0.24, 0.24, 0.24, 0.24, 0.24, 0.24</td>
<td>0.2, 0.24, 0.24, 0.24, 0.24, 0.24, 0.24, 0.24</td>
<td>0.2, 0.24, 0.24, 0.24, 0.24, 0.24, 0.24, 0.24</td>
</tr>
</tbody>
</table>

It can be easily checked that for all these cases, the corresponding obscurity matrix satisfies the requirement as stated in theorem 1.

In practice, usually not all two-patterns are required to test the stuck-open faults or delay faults of a combinational circuit. For example, the 2-input NAND shown in Fig 1 requires only 3 two-patterns instead of 12. Hence, a portion of the CA cycle is sufficient to check all stuck-open faults of the CUT. This implies that the testing time is usually much less than that in the worst case, since there is no need to run the CA through the entire cycle length.

Moreover, reduction in the hardware overhead (in terms of number of CA cells) may be achieved in such cases where only a subset of all two-patterns are required.

3 Experimental Results:

Experiments have been conducted with maximum length CA. In order to avoid large number of idle cycles which do not cover any of the two-patterns, loading of two different seeds has been also attempted. Simulation result shows that a portion of a whole CA cycles is required for generation of two-pattern test vectors. So for \( n \) input CUT, instead of \( 2n \) bit CA driven for \( (2^{2n} - 1) \) cycles, we can use only \( k(\leq 2^{2n}) \) cycles. Table 4 shows simulation results of some small real life circuits.
4 Conclusion:

Exhaustive two-pattern generator circuits are necessary to test arbitrary combinational circuits. In this paper we have arrived at the sufficient condition that has to be satisfied by such a circuit if exhaustive two-patterns are to be generated at predetermined cell positions. It has also been proved that any combination of Rule 90 and Rule 150 satisfies such criteria. To minimize loading overhead and circuit complexity the maximal length CA with least number of Rule 150 should be chosen. Finally, depending upon the test pattern requirement of the CUT, number of CA cells can also be reduced.

References


