Proof that akers’ algorithm for locally exhaustive testing gives minimum test sets of combinational circuits with up to four outputs

Hiroyuki Michinishi
Okayama University

Tokumi Yokohira
Okayama University

Takuji Okamoto
Okayama University
Proof that Akers' Algorithm for Locally Exhaustive Testing Gives Minimum Test Sets of Combinational Circuits with up to Four Outputs

Hiroyuki Michinishi, Tokumi Yokohira and Takuji Okamoto

Faculty of Engineering
Okayama University
3-1-1, Tushima-naka, Okayama-shi, 700 Japan

Abstract

In this paper, we prove that Akers' test generation algorithm for locally exhaustive testing gives a minimum test set (MLTS) for every combinational circuit (CUT) with up to four outputs. That is, we clarify that Akers' test pattern generator can generate an MLTS for such CUT.

1 Introduction

In built-in self-test of multiple output combinational circuits (CUTs), exhaustive testing is a simple testing method to raise fault coverage, whereas too many test patterns are necessary for the CUTs with large number of inputs.

In order to overcome the above problem, retaining the advantages of the exhaustive testing, the locally exhaustive testing[3,2], the pseudoeaxhaustive testing[3,4] and the verification testing[2] have been proposed. The difference among them is only in the naming, and the principal concepts are almost same. We use the first naming. In the locally exhaustive testing, if an output $y_i$ depends on $w_i$ inputs, a test set (LTS) is generated so that $2^{w_i}$ patterns are applied to them ($1 \leq i \leq m$; $m$ is the number of outputs). Many researchers, for example, Akers, Hiraishi, McCluskey, have proposed the algorithms to obtain LTSs. Using these algorithms, hardware generators for LTSs can be also obtained directly. These algorithms, however, do not guarantee to obtain a minimum test set (MLTS).

In general, an MLTS has more than or equal to $2^w$ elements, where $w$ is the maximum number of inputs on which any output depends. We have proposed an algorithm[6] to obtain an MLTS for every CUT with up to four outputs, and clarified that the number of test patterns is equal to $2^w$, independently of $n$, where $n$ is the number of inputs. It has not however been investigated how to construct a hardware generator for an MLTS. We call such a generator an MLTS generator.

In this paper, we show that Akers' algorithm gives an MLTS generator for every CUT with up to four outputs, that is, the algorithm gives an MLTS for such CUT.

In Section 2, the LTS, MLTS and a linear function are formally defined, and the relation between linear function and Akers' algorithm is described for the succeeding sections. In Section 3, two theorems closely related to linear function are established, and it is proved by the use of these theorems that Akers' algorithm gives an MLTS.

2 Akers' Algorithm

2.1 Definition of Minimum Locally Exhaustive Test Set

We shall consider a combinational circuit under test (CUT) having $n$ inputs $x_1, x_2, \ldots, x_n$, and $m$ outputs $y_1, y_2, \ldots, y_m$. Let a set $X$ be \(\{x_1, x_2, \ldots, x_n\}\), and let a set $X_i$ be \(\{x_i, \ldots, x_m\}\) (subset of $X$) when $y_i$ depends on $x_i, x_{i+1}, \ldots, x_m$ ($1 \leq i \leq m$, and $|X_i| = w_i$). It is assumed that $X_1 \cup X_2 \cup \ldots \cup X_m = X$ and the CUT remains combinational even if any fault occurs. A locally exhaustive test set, an LTS briefly, for the CUT is defined as follows[3].

[Definition 1] We call an $n$-dimensional vector $(x_1, x_2, \ldots, x_n)$ a test pattern. If a set $T$ of test patterns satisfies the following condition for $y_i (1 \leq i \leq m)$, then the set $T$ is an LTS.

Condition: The projection of $T$ onto $(x_1, x_2, \ldots, x_n)$ subspace corresponding to $X_i$ contains all $2^{w_i}$ distinct binary patterns.

Thus, an LTS is a set of test patterns which can exhaustively test each output of the CUT. If the number of test patterns is minimal, then the LTS is a minimum locally exhaustive test set, an MLTS briefly. Note that the number of test patterns in an MLTS is more than or equal to $2^w$ from the definition of the LTS, where $w = \max\{w_1, w_2, \ldots, w_m\}$.

2.2 Linear Function

In this section, we introduce the following definitions as preliminaries for the succeeding sections.

[Definition 2] When each of matrices $M_1, M_2, \ldots, M_k$ has the same number of row vectors, the concatenation of these matrices in this order, which is called a concatenated matrix $M$, is represented as follows[6].

\[
M \triangleq M_1 \times M_2 \times \cdots \times M_k
\]

[Definition 3] The dependence matrix $D_C$ for a CUT has $m$ row vectors and $n$ column vectors. The $i$th element is 1 iff the output $y_i$ depends on the input $x_j$, and is 0 otherwise.

Note that the weight of the $i$th row vector of a $D_C$ is equal to $w_i$, and the maximum row weight is equal to $w$.

[Definition 4] For $r \geq p > 1$, let $t_r$ be a column vector which has $2^p$ elements $(1 \leq p \leq r)$, and it is assumed that the concatenated matrix $t_r \times t_2 \times \cdots \times t_r$ has all binary $r$-dimensional row vectors. Then, the set \{ $t_1, t_2, \ldots, t_r$ \} is called a base set.
Thus, the assignment of an element $f_0$ for $X_i$ (recall that $\{f_1, f_2, \ldots, f_r\}$ has all binary $q$-dimensional row vectors.

[Definition 7] For a given linear function set $S (\triangleq \{f_1, f_2, \ldots, f_r\})$, the set of all linear combinations of $f_1, f_2, \ldots, f_r$ is represented by $F(S)$ or $F(f_1, f_2, \ldots, f_r)$.

Note that, a given linear function set $\{f_1, f_2, \ldots, f_r\}$ is $q$-independent if $F(f_1, f_2, \ldots, f_r)$ has $2^r - 1$ elements. Thus, by constructing $F(\cdots)$, we can examine whether a given linear function set is $q$-independent or not.

[Definition 8] For two distinct linear functions $f (\triangleq k_1 t_1 \oplus k_2 t_2 \oplus \cdots \oplus k_n t_n)$ and $f' (\triangleq k'_1 t_1 \oplus k'_2 t_2 \oplus \cdots \oplus k'_n t_n)$, if $\sum_{m=1}^n k_m 2^{m-1} < \sum_{m=1}^n k'_m 2^{m-1}$, we call that $f$ is smaller than $f'$.

If, for example, let $f \triangleq t_1 \oplus t_2$ and $f' \triangleq t_1 \oplus t_3$, then $f$ is smaller than $f'$.

2.3 Akers' Linear Function Assignment Algorithm

Akers’ test pattern generator is based on linear function assignment described below.

[Definition 9] Let $G$ be a set of $u$ linear functions $f_1, f_2, \ldots, f_u (w \leq u \leq n)$, and assume that there exists such a mapping $g$ from $X$ onto $G$ that satisfies the following condition for $X_i$ (recall that $X_i \triangleq \{x_1, x_2, \ldots, x_{w_i}\}$), then we call that the CUT or the corresponding dependence matrix $D_C$ is $r$-assignable.

Condition: If $g(x_j) = f_j (1 \leq j \leq w_i)$, then the set $\{f_1, f_2, \ldots, f_{w_i}\}$ is $w_i$-independent.

If $f_i = g(x_j)$, then we call that the linear function $f_i$ is assigned to the input $x_j$. Note that, if a CUT is $r$-assignable, then $r$ is greater than or equal to $w_i$.

Suppose a CUT whose dependence matrix is shown in Figure 1(a). If $t_4, t_5, t_6, t_3$ and $t_1 \oplus t_2$ are assigned to $x_1$, $x_2$, $x_3$, $x_4$, and $x_5$, respectively, then the condition above is satisfied. Figure 1(b) shows $t_4 \equiv t_1 \equiv t_2$ and $t_3 \equiv t_1 \oplus t_2$. From the definition 6, Figure 1(b) is a matrix representation of an LTS for the CUT.

Each row vector of the matrix constructed with $t_1, t_2, \ldots, t_n$ can be easily generated by a maximum sequence generator. Thus, if a CUT is $r$-assignable, then a test pattern generator constructed with a maximum sequence generator and EXOR gates can be easily obtained. For example, Figure 1(b) can be generated with a test pattern generator shown in Figure 2.

For a given $D_C$, Akers’ algorithm assigns linear functions as follows:

[Akers’ Assignment Algorithm]

(A-1) $r = w_i$.

(A-2) Select an arbitrary input $x_j$ to which a linear function is not assigned, and find all output $y_{i1}$, $y_{i2}$, $\ldots$, $y_{i\ell_i}$ which depend on $x_j$. Next, for each output $y_{i\ell_i}$ (1 \leq \ell_i \leq \ell), find all inputs to which linear functions have been already assigned, and construct a set $E_{i\ell_i}$ of such linear functions (for an output $y_{i\ell_i}$, if $y_{i\ell_i}$ does not have an input to which a linear function has been already assigned, then $E_{i\ell_i} = \phi$).

(A-3) Construct an set $S^1$ according to the following equation.

$S^1 \triangleq F(L_{i1}^1) \cup F(L_{i2}^1) \cup \cdots \cup F(L_{i\ell_i}^1)$.

Next, construct $F(T^*)$, where $T^* \triangleq \{t_1, t_2, \ldots, t_r\}$. If $|S^1| < |F(T^*)|$, then execute the following procedure (A-3.1.1), otherwise, execute the following procedure (A-3.2.2).

(A-3.1.1) Assign the smallest linear function in the set $S^1$ to $x_j$.

(A-3.2.2) Assign $t_{r+1}$ to $x_j$, and increase the value of $r$ by 1.
3 Proof that Akers’ Algorithm Gives an MLTS

The basic problem with respect to linear function assignment is to find such a mapping \( g \) that the value of \( r \) is minimum, because the smaller the value of \( r \) is, the smaller the number of test patterns is. Unfortunately, the problem is an NP-complete one[2]. Though Akers’ algorithm is straightforward and time-effective, it does not guarantee to obtain the minimum value of \( r \).

In this section, we prove that the minimum value of \( r \) can be obtained from Akers’ algorithm and is always equal to the value of \( w \) for every CUT with up to four outputs. It is trivial that, if any CUT with four outputs is \( w \)-assignable, then every CUT with less than four outputs is also \( w \)-assignable. Thus, we prove only for four outputs.

Without loss of generality, it is assumed that a given dependence matrix \( D_c \) has the following properties (see Figure 3).

[A]ssumption-1 The weight of the row vector which corresponds to the output \( y_i \) is \( w (w_1 = w) \), and \( X_1 = \{ x_1, x_2, \ldots, x_w \} \).

[B]ssumption-2 If \( D_c \) has \( u \) columns whose weight are four (\( w \leq u \)), these column vectors are located in \( u \) successive column vectors starting with first column vector.

And without loss of generality, we assume that the arbitrary selection in the procedures (A-2) and (A-3.1) of Akers’ algorithm are determined as follows:

[A]ssumption-3 In the procedure (A-2), \( y_1 \) is selected as \( y_i \), and \( t_1, t_2, \ldots, t_w \) are assigned to \( x_1, x_2, \ldots, x_w \), respectively.

[A]ssumption-4 In the \( j \)-th procedure (A-3.1), \( x_{w+j} \), is selected as \( x_j \) (\( 1 \leq j \leq n - w \)). That is, a linear function is assigned to each of \( x_{w+j}, x_{w+j}, \ldots, x_n \) in this order.

\[ \begin{array}{cccccccc}
1 & 1 & 1 & 1 & 0 & \cdots & 0 & 0 \\
1 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\
1 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\
\end{array} \]

Figure 3 General Form of Dependence Matrix.

Under the assumptions above, it is proved that \( |S^{w+j}| < |F(T^w)| \) for \( w \) and \( j \) (\( 1 \leq j \leq n - w \)) in the \( j \)-th visit of procedure (A-3.2), then a given \( D_c \) with the maximum row weight \( w \) becomes \( w \)-assignable, where \( T^w = \{ t_1, t_2, \ldots, t_w \} \). So, we prove that \( |S^{w+j}| < |F(T^w)| \) for the three cases, \( w = 1, w = 2 \) and \( w = 3 \). The proof for each case is performed by induction with respect to \( j \).

In this section, two theorems are established, and the proof is done using the theorems.

3.1 Theorems for the Proof

In the discussions below, we simply represent a column vector and a row vector of a given \( D_c \) by a column vector and a row vector, and we represent the column vector which corresponds to \( x_{w+j} \), by \( \{ 0, a_2, a_3, a_4 \}^T \), where \( v^T \) represents the transpose of a row vector \( v \). Without loss of generality, we assume that \( (a_2, a_3, a_4) = (1, 0, 0) \) or \( (1, 1, 0) \) or \( (1, 1, 1) \) (note that \( (a_2, a_3, a_4) \neq (0, 0, 0) \) since it is assumed that \( X = X_1 \cup X_2 \cup \cdots \cup X_m \).

Let \( (b_1, b_2, b_3, b_4)^T \) be the \( w \)th column vector (which corresponds to \( x_w \)). If \( (b_1, b_2, b_3) = (1, 1, 1) \), then all elements of a given \( D_c \) are is from Assumption-2, i.e., \( w_1 = w_2 = w_3 = w_4 = w = n \). In this case, it is trivial that a given \( C \) is \( w \)-assignable (the procedures (A-3.1) and (A-3.2) of Akers’ algorithm are not executed). Thus, in the discussions below, we assume that \( (b_1, b_2, b_3, b_4) \neq (1, 1, 1) \).

[Theorem 1] For \( w \) and \( j \) (\( 1 \leq j \leq n - w \)), the following property holds.

[Property-1] Assume that \( (a_2, a_3, a_4) = (1, 0, 0) \) or \( (1, 1, 0) \). And consider a matrix constructed by removing the \((w + j)\)th to nth column vectors from a given \( D_c \) as a new dependence matrix \( D_c' \) (note that the maximum row weights of \( D_c' \) is equal to that of \( D_c \) from the general form of dependence matrix).

If \( D_c' \) is \( w \)-assignable, then \( |S^{w+j}| < |F(T^w)| \).

[Proof of Theorem 1] If \( (a_2, a_3, a_4) = (1, 0, 0) \), then \( S^{w+j} = F(L_2^{w+j}) \). Since \( D_c' \) is \( w \)-assignable, \( L_2^{w+j} \) is \( w \)-independent, and consequently, \( |F(L_2^{w+j})| = 2^n - 1 \), where \( q_2 = 1 \). On the other hand, since \( a_2 = 1 \), \( q_2 \leq w - 1 \) (otherwise, a contradiction that \( w_2 \) is larger than \( w \) occurs). Thus, the following relation holds.

\[ |S^{w+j}| = |F(L_2^{w+j})| = 2^n - 1 - 2^{w-1} - 1 = |F(T^w)| (2) \]

If \( a_2, a_3, a_4 = (1, 1, 0) \), then \( S^{w+j} = F(L_3^{w+j}) \cup F(L_4^{w+j}) \). Since \( a_3 = 1 \), \( q_2 \leq w - 1 \), where \( q_3 = 1 \). Thus, the following relation holds.

\[ |S^{w+j}| = |F(L_3^{w+j})| + |F(L_4^{w+j})| \leq |F(L_3^{w+j})| + |F(L_4^{w+j})| \leq 2^n - 1 + 2^{w-1} - 1 \]

\[ = 2^n - 1 - 1 = |F(T^w)|. \]

[Theorem 2] Let two linear function sets \( L \) and \( L' \) be \( \{ f_1, f_2, \ldots, f_{w-1} \} \) and \( \{ f_1, f_2, \ldots, f_{w-1}, f_w \} \), respectively, where \( u \leq w - 1 \), and assume that \( L \) and \( L' \) are \((w-1)\)-independent and \( u \)-independent, respectively. Then the following equation holds.

\[ |F(L) \cap F(L')| = \begin{cases} 2^{w-1} - 1 & (F(L) \supseteq F(L')) \\ 2^{w-1} - 1 & (F(L) \supseteq F(L')) \end{cases} \]

[Definition 10] Let a linear function set \( L \) be \( \{ f_1, f_2, \ldots, f_{w-1} \} \), and assume that a linear function \( f \) is not an element of \( L \). We represent the set \( \{ f \otimes f_1, f \otimes f_2, \ldots, f \otimes f_{w-1} \} \) by \( f \otimes L \).

[Proof of Theorem 2] It is trivial for the case that \( F(L) \supseteq F(L') \). Thus, we prove for the case that \( F(L) \supseteq F(L') \). If it is assumed that all elements of \( L' \) are elements of \( F(L) \), then \( F(L) = F(L') \). Thus, in the case that \( F(L) \supseteq F(L') \), there exists such an element of \( L' \) that is not an element of \( F(L) \). Without loss of generality, let \( \{ f_1', f_2', \ldots, f_{w-1}' \} \) be a set of such elements that are not included in \( F(L) \). We prove the following three cases.

Case-1: \( u = 1 \). Since \( L' = \{ f_w \} \), \( F(L') = \{ f_w \} \) . On the other hand, \( f_w \not\in F(L) \). Thus, \( F(L) \cap F(L') = \emptyset \). Therefore, \( |F(L) \cap F(L')| = 2^n - 1 \).

Case-2: \( u \geq 2 \) and \( q = u \) (see Figure 4(a)).

16
(i) \( F(L') \) is represented as follows:

\[
F(L') = F(f_1', f_2', \ldots, f_{u-1}') \cup \{f_u'\} \cup (f_u' \oplus F(f_1', f_2', \ldots, f_{u-1}')).
\]

(ii) The following equation holds.

\[
F(L) \cap (f_u' \oplus F(f_1', f_2', \ldots, f_{u-1}')) = \phi.
\]

(iii) Since \( f_1', f_2', \ldots, f_{u-1}' \) are elements of \( F(L) \), \( F(f_1', f_2', \ldots, f_{u-1}') \subseteq F(L) \).

(iv) \( f_u' \) is not an element of \( F(L) \).

(v) From (i) and (iv), the following equation holds.

\[
F(L) \cap F(L') = F(f_1', f_2', \ldots, f_{u-1}').
\]

The set \( \{f_1', f_2', \ldots, f_{u-1}'\} \) is \( (u-1) \)-independent.

Thus, \( |F(L) \cap F(L')| = 2^{u-1} - 1 \).

---

Case 3: \( u \geq 2 \) and \( 1 \leq q \leq u - 1 \) (see Figure 4(b)).

Since \( f_q' \notin F(L) \) (1 \( \leq q \leq u \)), \( f_q' \) is an element of \( \{f_u\} \cup (f_u \oplus F(L')) \), where \( f_u \) is such a linear function that the set \( \{f_1, f_2, \ldots, f_{u-1}, f_u\} \) is \( u \)-independent. Thus, \( f_q' \) is represented as follows:

\[
f_q' = f_u \oplus k_1^q f_1 \oplus k_2^q f_2 \oplus \cdots \oplus k_{u-1}^q f_{u-1}.
\]

where \( k_1^q, k_2^q, \ldots, k_{u-1}^q \) are defined by \( \mu \) such that \( \mu(q-1) \). Thus, for \( \forall q' (1 \leq q' \leq u) \), the following equation holds:

\[
f_q' \oplus f_q' = k_1^q f_1 \oplus k_2^q f_2 \oplus \cdots \oplus k_{u-1}^q f_{u-1}.
\]

On the other hand, let \( L'' \) be \( \{f_1', f_2', \ldots, f_{u-1}'\}, f_q', f_q' \oplus f_q', f_q' \oplus f_{q+1}', \ldots, f_q' \oplus f_u' \), then \( L'' \) is \( u \)-independent, and subset of \( F(L') \). Therefore, \( F(L) \subseteq F(L') \).

Therefore, relation between \( L'' \) and \( L \) in Case 2. Thus, \( |F(L) \cap F(L')| = 2^{u-1} - 1 \). Consequently, \( |F(L) \cap F(L')| = 2^{u-1} - 1 \).

3.2 Proof that \( |S^{w+1}| < |F(T^{w})| \)

For \( w = 1 \), we prove by induction with respect to \( j_1 \).

(Basis Step : \( w = 1 \)) From Assumption 3, the assumptions of Property 1 are satisfied. From Theorem 1, the proof is trivial for the case that \( (a_2, a_3, a_4) = (1, 1, 0) \) or \( (1, 1, 0) \). If \( (a_2, a_3, a_4) = (1, 1, 1) \), then \( b_2 = b_3 = b_4 = 0 \), since \( w = 1 \) (see Figure 3(a)). Therefore, each of \( L_2', L_3', L_4' \) is an empty set. Thus, \( |S^1| = |F(L_2')| \cup |F(L_3')| \cup |F(L_4')| \).

(Induction Step : \( w = 2 \)) If \( (a_2, a_3, a_4) = (1, 1, 0) \) or \( (1, 1, 0) \), then in the case of \( w = 1 \) the discussion in the basis step similarly holds. If \( (a_2, a_3, a_4) = (1, 1, 1) \), then the general form of \( DC \) becomes as shown in Figure 3(b). All elements of \( X \) are \( 0 \). But this is contradictory to the assumption that \( X = X_1 \cup X_2 \cup \ldots \cup X_m \). In other words, if \( j_1 > 1 \), then there does not exist such a case that \( (a_2, a_3, a_4) = (1, 1, 1) \).

For \( w = 2 \), we prove by induction with respect to \( j_1 \).

(Basis Step : \( w = 2 \)) If \( (a_2, a_3, a_4) = (1, 0, 0) \) or \( (1, 0, 0) \), then the proof is trivial from Theorem 1. The proof for the case that \( (a_2, a_3, a_4) = (1, 1, 1) \) is as follows:

If \( c_2, c_3, c_4 \) are defined as shown in Figure 6(a), then \( (c_2, b_i) \neq (1, 1) \), since \( w = 2 \). Thus, \( L_2^2 \leq 1, L_3^2 \leq 1 \) and \( L_4^2 \leq 1 \) and it is trivial that \( L_{1i} = \phi \) for \( i \in I \) or \( i \in I' \) or \( i \in I'' \). Therefore, the following relation holds:

\[
|S^2| = |F(L_2')| \cup |F(L_3')| \cup |F(L_4')|
\]

\[
|F(L_2')| \cup |F(L_3')| \leq |F(L_2')| + |F(L_3')| \leq 2 < 3 = |F(T^2)|.
\]

Similarly, we have \( |S^{2^w}| < |F(T^{w})| \) for the case that \( y_2 \) or \( y_4 \) does not depend on each of inputs \( x_1, x_2, \ldots, x_{2w}, x_{2w+1} \).

Thus, we assume that each of outputs \( y_2, y_3, y_4 \) depends on one of \( x_1, x_2, \ldots, x_{2w}, x_{2w+1} \) (this situation can occur only when \( j_1 > n - w \), since \( w = 2 \)). Let \( x_{2w}, x_{2w+1} \) be such inputs for \( y_2, y_3, y_4 \), respectively. If \( x_2, x_3, x_4 \) are identical inputs, then the relation as \( (11) \) holds. Similarly, we have \( |S^{2^w}| < |F(T^{w})| \) for the case that \( x_2, x_3, x_4 \) are identical inputs.

Thus, in the discussions below, we assume that \( x_{2w}, x_{2w+1} \) are different each other, and without loss of generality, we assume that \( \alpha = \beta < \gamma \).

Figure 5 shows that the general form of \( DC \) in case that \( w = 1 \) and \( (a_2, a_3, a_4) = (1, 1, 1) \).
From Assumption-1 and \( w = 2 \), all elements in a shadow area of the first row vector are 0s. And from \( w = 2 \), all elements in shadow areas of each of the second, third and fourth row vectors are 0s.

(i) Let \( f_x, f_y \) and \( f_z \) be linear functions which are assigned to \( x_1, x_2 \) and \( z_1 \), respectively. The first, second and third rows of the \( i \)-th column vector are 0s. In the \((\gamma - w)\)-th visit of \((A, 3, 1)\), i.e., in the assignment to \( x_1 \), therefore, \( S^T = F(L^x) = \phi \). Since \( t_i \) is the smallest linear function of \( F(T^x) \) \( \uparrow \{ t_1, t_2, t_3 \} \), therefore, \( t_i \) is assigned to \( x_1 \), i.e., \( f_x = t_i \).

(ii) If \( \beta = 1 \), then \( \alpha_0 = 1 \) from Assumption-1 and \( w = 2 \), i.e., \( z_{a_0} \) and \( x_{a_0} \) are identical to \( x_1 \) and \( x_2 \), respectively. Thus, \( f_x = t_i \). From (i), therefore, \( L^{22; i} = L^{22} \) in the assignment to \( x_{22} \). Thus, the same relation as (11) holds.

(iii) If \( \beta = 0 \), then the first, second and fourth rows of the \( i \)-th column vector are 0s. Thus, \( S^T = F(L^x) = \phi \). Therefore, \( f_x = t_i \). From (i), therefore, \( L^{22} = L^{22; i} \) in the assignment to \( x_{22} \). Thus, we have \( |F(T^x)| < |F(T^{22})| \) by replacing \( L^{22; i} \) in (11) with \( L^{22} \).

We assume that any \( D_2 \) with the maximum row weight \((w - 1)\) is \((w - 1)\)-assignable, and we prove that \( |S^{w^* + 1}| < |F(T^{w^* + 1})| \) for \( \uparrow \{ \gamma \} \) in any \( D_2 \) with the maximum row weight \( w \).

**[Basis Step] \( w \geq 3 \)**

If \( (a_2, a_3, a_4) = (1, 0, 0) \) or \((1, 1, 1)\), then the proof is trivial from Theorem 1. The proof for the case that \( (a_2, a_3, a_4) = (1, 1, 1) \) is as follows:

Let \( q_i \triangleq |L^{w^* + 1}| \) \( (2 \leq i \leq 4; \ w \geq w - 1) \). If \( F(L^{w^* + 1}) \subseteq F(L^{w^* + 1}) \) for \( i \); \( i \neq j \) and \( i \neq k \); \( \left. \uparrow \{ i \} \right. \) of the following relation holds.

\[
|S^{w^* + 1}| = |F(L^{w^* + 1})| = |F(L^{w^* + 1})| \\
\leq 3q_i - 1 + 2w - 1 \leq 2w - 1 + 2w - 1 - 1 < 2w - 1 - 1 = |F(T^{w^*})|.
\]

where \( i \neq 1 \) and \( i \neq 2 \). If \( q_i < 0 \) \( (2 \leq i \leq 4; j = 0) \), then the same relation as (12) holds. Thus, in the discussions below, we assume that \( L^{w^* + 1} \notin F(L^{w^* + 1}) \) for \( \left. \uparrow \{ i \} \right. \) and \( \left. \leftarrow \{ i \} \right. \) \( (i \neq j \) and \( i \neq k \), and assume that \( q_i \geq 1 \) for \( \left. \uparrow \{ i \} \right. \).

Without loss of generality, we assume that \( w - 1 \geq q_i \) \( \geq q_2 \) \( \geq q_4 \), and prove the following four cases.

**Case 1:** \( w - 2 \geq q_i \geq q_2 \geq q_4 \)

\[
|S^{w^* + 1}| = |F(L^{w^* + 1})| + |F(L^{w^* + 1})| + |F(L^{w^* + 1})| \\
= |F(L^{w^* + 1})| + |F(L^{w^* + 1})| + |F(L^{w^* + 1})| \\
= 2w - 2 + 2w - 1 + 2w - 1 - 1 \leq 2w - 1 - 1 = |F(T^{w^*})|.
\]

**Case 2:** \( w - 1 = q_i \), \( w - 2 \geq q_2 \) \( \geq q_4 \)

\[
|S^{w^* + 1}| = |F(L^{w^* + 1})| + |F(L^{w^* + 1})| + |F(L^{w^* + 1})| \\
= |F(L^{w^* + 1})| + |F(L^{w^* + 1})| + |F(L^{w^* + 1})| \\
= 2w - 2 + 2w - 1 + 2w - 1 - 1 \leq 2w - 1 - 1 = |F(T^{w^*})|.
\]

**Case 3:** \( w - 1 = q_i \), \( q_2 \geq q_2 \geq q_4 \)

\[
|S^{w^* + 1}| = |F(L^{w^* + 1})| + |F(L^{w^* + 1})| + |F(L^{w^* + 1})| \\
= |F(L^{w^* + 1})| + |F(L^{w^* + 1})| + |F(L^{w^* + 1})| \\
= 2w - 2 + 2w - 1 + 2w - 1 - 1 \leq 2w - 1 - 1 = |F(T^{w^*})|.
\]

**Case 4:** \( q_i \geq q_2 \) \( \geq q_4 \) \( w - 2 \geq q_2 \) \( \geq q_4 \)

\[
|S^{w^* + 1}| = |F(L^{w^* + 1})| + |F(L^{w^* + 1})| + |F(L^{w^* + 1})| \\
= |F(L^{w^* + 1})| + |F(L^{w^* + 1})| + |F(L^{w^* + 1})| \\
= 2w - 2 + 2w - 1 + 2w - 1 - 1 \leq 2w - 1 - 1 = |F(T^{w^*})|.
\]
or the basis step hold by replacing \( \omega \) with \( \omega + 1 \).

Thus, in the discussions below, we assume that \( q_2 = q_3 = q_4 = w - 1 \), where \( q_i \neq 0 \) if \( F(L_1^{\omega+ji}) \neq F(L_2^{\omega+ji}) \), and \( F(L_3^{\omega+ji}) \neq F(L_2^{\omega+ji}) \) and \( F(L_2^{\omega+ji}) \neq F(L_4^{\omega+ji}) \) (see Figure 7(b)). Note that this case can occur when \( j_1 = 1 \).

Thus, \( (w, \omega + 1) \) appears in the expression of any linear function of \( L_1^{\omega+ji} \), and consequently, \( t_w \) never appears in the expression of any linear function of \( L_4^{\omega+ji} \), and consequently, \( t_w \) never appears in the expression of any linear function of \( F(L_4^{\omega+ji}) \).

Therefore, \( F(L_1^{\omega+ji}) \cap F(L_2^{\omega+ji}) \cap F(L_3^{\omega+ji}) \cap F(L_4^{\omega+ji}) \). Thus, from Theorem 2, \( 2^{w-2} - 1 \leq |F(L_1^{\omega+ji}) \cap F(L_2^{\omega+ji}) \cap F(L_3^{\omega+ji}) \cap F(L_4^{\omega+ji})| \). On the other hand, it is trivial that an equation which is obtained by replacing \( w+1 \) in (15) with \( w+1 \) holds. Thus, the following relation holds:

\[
|S^{\omega+ji}| = 2^w + 2^{w-1} - 2^{w-2} - 1 = 2^w - 1 = |F(T^w)|.
\]

4 Conclusion

In this paper, we showed that a hardware MLTS generator for every CUT with up to four outputs can be constructed using a maximum sequence generator with \( w \) stages and EXOR gates, by giving proof that Akers’ algorithm gives an MLTS for such CUT.

We can easily prove that there does not exist such a generator for some CUT with more than five outputs. It is however an open problem whether there exists such a generator for every CUT with five outputs or not.

References


