Engineering

Industrial & Management Engineering fields

Okayama University *Year* 1993

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 algorithmfor the 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[1,2], the pseudoexhaustive testing[3,4] and the verification testing[5] 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, that 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 $x_1, x_2, \cdots, x_n$, and m outputs $y_1, y_2, \cdots, y_m$. Let a set X be $\{x_1, x_2, \cdots, x_n\}$, and let a set X_i be $\{x_1^i, x_2^i, \cdots, x_{w_i}^i\}$ ($\subseteq X$) when y_i depends on $x_1^i, x_2^i, \cdots, x_{w_i}^i$ ($1 \leq i \leq m$, and $|X_i| = w_i$). It is assumed that $X_1 \cup X_2 \cup \cdots \cup X_m = X$ 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[5].

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

Condition : The projection of T onto $(x_1^i, x_2^i, \cdots, x_{w_i}^i)$ 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 $w \triangleq max\{w_1, w_2, \cdots, w_m\}$.

2.2 Linear Function

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

[Definition 2] When each of matrices $M_1, M_2, \cdots, M_k$ 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]:

$$M \triangleq M_1 \bowtie M_2 \bowtie \cdots \bowtie M_k. \quad ■$$

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

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

[Definition 4] For $\forall r$ ($r \geq 1$), let t_p be a column vector which has 2^r elements ($1 \leq p \leq r$), and it is assumed that the concatenated matrix $t_1 \bowtie t_2 \bowtie \cdots \bowtie t_r$ has all binary r-dimensional row vectors. Then, the set $\{t_1, t_2, \cdots, t_r\}$ is called *a base set*. ■

[Definition 5] A linear combination $k_1t_1 \oplus k_2t_2 \oplus \cdots \oplus k_rt_r$ is called *a linear function*, where $k_1, k_2, \cdots, k_r \in \{0, 1\}$ and $(k_1, k_2, \cdots, k_r) \neq (0, 0, \cdots, 0)$. ■
Note that there exists $2^r - 1$ linear functions.

In the discussions below, we implicitly assume that a base set is T^r ($\triangleq \{t_1, t_2, \cdots, t_r\}$), and that linear functions are linear combinations of $t_1, t_2, \cdots, t_r$.

[Definition 6] The set of q distinct linear functions $f_1, f_2, \cdots, f_q$ is called *q-independent* if $f_1 \bowtie f_2 \bowtie \cdots \bowtie f_q$ has all binary q-dimensional row vectors. ■

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

Note that, a given linear function set $\{f_1, f_2, \cdots, f_q\}$ is q-independent iff $F(f_1, f_2, \cdots, f_q)$ has $2^q - 1$ elements[1]. 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 ($\triangleq k_1t_1 \oplus k_2t_2 \oplus \cdots \oplus k_rt_r$) and f' ($\triangleq k'_1t_1 \oplus k'_2t_2 \oplus \cdots \oplus k'_rt_r$), if $\sum_{p=1}^{r} k_p2^{p-1} < \sum_{p=1}^{r} k'_p2^{p-1}$, then we call that f is smaller than f'. ■

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 $f_1, f_2, \cdots, f_u$ ($w \leq u \leq n$), and assume that there exists such a mapping g from X onto G that satisfies the following condition for $^\forall X_i$ (recall that $X_i \triangleq \{x_1^i, x_2^i, \cdots, x_{w_i}^i\}$), then we call that the CUT or the corresponding dependence matrix D_C is *r-assignable*.

Condition: If $g(x_j^i) = f_j^i$ ($1 \leq j \leq w_i$), then the set $\{f_1^i, f_2^i, \cdots, f_{w_i}^i\}$ 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.

Suppose a CUT whose dependence matrix is shown in Figure 1(a). If t_4, t_1, t_2, 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 $t_4 \bowtie t_1 \bowtie t_2 \bowtie t_3 \bowtie (t_1 \oplus t_2)$. From the definition 6, Figure 1(b) is therefore a matrix representation of an LTS for the CUT.

Each row vector of the matrix constructed with t_1, t_2, $\cdots$, t_r 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) $r = w$.

(A-2) Select such an arbitrary output y_i that the weight of the corresponding row vector in the D_C is equal to w, and assign t_j to each input x_j^i ($1 \leq j \leq w_i = w$).

	x_1 (t_4)	x_2 (t_1)	x_3 (t_2)	x_4 (t_3)	x_5 ($t_1 \oplus t_2$)
y_1	1	1	0	1	0
y_2	0	0	1	1	1
y_3	1	0	1	1	0
y_4	0	1	1	1	0
y_5	0	1	0	0	1
y_6	1	0	0	1	1

(a) Dependence Matrix

x_1	x_2	x_3	x_4	x_5
0	0	0	0	0
1	0	0	0	0
0	0	0	1	0
1	0	0	1	0
0	0	1	0	1
1	0	1	0	1
0	0	1	1	1
1	0	1	1	1
0	1	0	0	1
1	1	0	0	1
0	1	0	1	1
1	1	0	1	1
0	1	1	0	0
1	1	1	0	0
0	1	1	1	0
1	1	1	1	0
t_4	t_1	t_2	t_3	$t_1 \oplus t_2$

(b) LTS

Figure 1 Relation between Dependence Matrix and LTS.

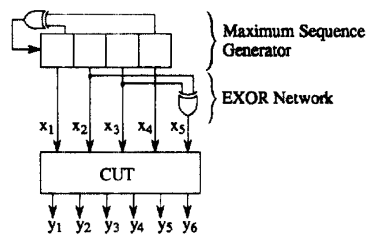


Figure 2 Test Pattern Generator for the LTS shown in Figure 1(b).

(A-3) Repeat the following procedures (A-3.1) and (A-3.2) until a linear function is assigned to every input.

(A-3.1) Select an arbitrary input x_j to which a linear function is not assigned, and find all output $y_{i_1}^j$, $y_{i_2}^j$, $\cdots$, $y_{i_c}^j$ which depend on x_j. Next, for each output $y_{i_v}^j$ ($1 \leq v \leq c$), find all inputs to which linear functions have been already assigned, and construct a set $L_{i_v}^j$ of such linear functions (for an output $y_{i_v}^j$, if $y_{i_v}^j$ does not have an input to which a linear function has been already assigned, then $L_{i_v}^j = \phi$).

(A-3.2) Construct an set S^j according to the following equation.

$$S^j \triangleq F(L_{i_1}^j) \cup F(L_{i_2}^j) \cup \cdots \cup F(L_{i_c}^j). \quad (1)$$

Next, construct $F(T^r)$, where $T^r \triangleq \{t_1, t_2, \cdots, t_r\}$. If $|S^j| < |F(T^r)|$, then execute the following procedure (A-3.2.1), otherwise, execute the following procedure (A-3.2.2).

(A-3.2.1) Assign the smallest linear function in the set $\overline{S^j}$ to x_j.

(A-3.2.2) Assign t_{r+1} to x_j, and increase the value of r by 1. ■

Thus, if $L_{i_v}^j = \{f_1^{i_v}, f_2^{i_v}, \cdots, f_{q_{i_v}}^{i_v}\}$, where $q_{i_v} \triangleq |L_{i_v}^j|$, then the procedure (A-3.2) assigns such a linear function f that $\{f_1^{i_1}, f_2^{i_1}, \cdots, f_{q_{i_1}}^{i_1}, f\}$, $\{f_1^{i_2}, f_2^{i_2}, \cdots, f_{q_{i_2}}^{i_2}, f\}$, $\cdots$, $\{f_1^{i_c}, f_2^{i_c}, \cdots, f_{q_{i_c}}^{i_c}, f\}$ become (q_{i_1}+1)-independent, (q_{i_2}+1)-

independent, $\cdots$, $(q_{i_c}+1)$-independent, respectively.

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).

[Assumption-1] The weight of the row vector which corresponds to the output y_1 is w ($w_1 = w$), and $X_1 = \{x_1, x_2, \cdots, x_w\}$. ■

[Assumption-2] If D_C has u column vectors whose weight are four ($u \leq w$), 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:

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

[Assumption-4] In the j_1th procedure (A-3.1), x_{w+j_1} is selected as x_j ($1 \leq j_1 \leq n-w$). That is, a linear function is assigned to each of $x_{w+1}, x_{w+2}, \cdots, x_n$ in this order. ■

	x_1	x_2	$\cdots$	x_{w-1}	x_w	x_{w+1}		x_{w+j_1}			x_n
y_1	1	1	$\cdots$	1	1	0	$\cdots$	0	0	$\cdots$	0
y_2					b_2			a_2			
y_3					b_3			a_3			
y_4					b_4			a_4			

Figure 3 General Form of Dependence Matrix.

Under the assumptions above, if it is proved that $|S^{w+j_1}| < |F(T^w)|$ for $^\forall w$ and $^\forall j_1$ ($1 \leq j_1 \leq n-w$) in the j_1th visit of procedure (A-3.2), then a given D_C with the maximum row weight w becomes w-assignable, where $T^w \triangleq \{t_1, t_2, \cdots, t_w\}$. So, we prove that $|S^{w+j_1}| < |F(T^w)|$ for the three cases, $w = 1$, $w = 2$ and $w \geq 3$. The proof for each case is performed by induction with respect to j_1.

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_1} 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 $(1, b_2, b_3, b_4)^T$ be the wth column vector (which corresponds to x_w). If $(b_2, b_3, b_4) = (1,1,1)$, then all elements of a given D_C are 1s from Assumption-2, i.e., $w_1 = w_2 = w_3 = w_4 = w = n$. In this case, it is trivial that a given CUT 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 $(b_2, b_3, b_4) \neq (1,1,1)$.

[Theorem 1] For $^\forall w$ and $^\forall j_1$ ($1 \leq j_1 \leq n-w$), 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 $(w+j_1)$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 $|S^{w+j_1}| < |F(T^w)|$. ■

[Proof of Theorem 1] If $(a_2, a_3, a_4) = (1,0,0)$, then $S^{w+j_1} = F(L_2^{w+j_1})$. Since D'_C is w-assignable, $L_2^{w+j_1}$ is q_2-independent, and consequently, $|F(L_2^{w+j_1})| = 2^{q_2} - 1$, where $q_2 \triangleq |L_2^{w+j_1}|$. 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.

$$|S^{w+j_1}| = |F(L_2^{w+j_1})| = 2^{q_2} - 1 < 2^w - 1 = |F(T^w)|. \quad (2)$$

If $(a_2, a_3, a_4) = (1, 1, 0)$, then $S^{w+j_1} = F(L_2^{w+j_1}) \cup F(L_3^{w+j_1})$. Since $a_3 = 1$, $q_3 \leq w - 1$, where $q_3 \triangleq |L_3^{w+j_1}|$. Thus, the following relation holds.

$$\begin{aligned} |S^{w+j_1}| &= |F(L_2^{w+j_1}) \cup F(L_3^{w+j_1})| \\ &\leq |F(L_2^{w+j_1})| + |F(L_3^{w+j_1})| \\ &= 2^{q_2} - 1 + 2^{q_3} - 1 \leq 2^{w-1} - 1 + 2^{w-1} - 1 \\ &< 2^w - 1 = |F(T^w)|. \end{aligned} \quad (3)$$

■

[Theorem 2] Let two linear function sets L and L' be $\{ f_1, f_2, \cdots, f_{w-1} \}$ and $\{ f'_1, f'_2, \cdots, f'_u \}$, 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.

$$|F(L) \cap F(L')| = \begin{cases} 2^u - 1 & (F(L) \supseteq F(L')), \\ 2^{u-1} - 1 & (F(L) \not\supseteq F(L')). \end{cases} \quad (4)$$

■

[Definition 10] Let a linear function set L be $\{ f_1, f_2, \cdots, f_q \}$, and assume that a linear function f is not an element of L. We represent the set $\{f \oplus f_1, f \oplus f_2, \cdots, f \oplus f_q\}$ by $f \oplus L$. ■

[Proof of Theorem 2] It is trivial for the case that $F(L) \supseteq F(L')$. Thus, we prove for the case that $F(L) \not\supseteq F(L')$. If it is assumed that all elements of L' are elements of $F(L)$, then $F(L) \supseteq F(L')$. Thus, in the case that $F(L) \not\supseteq F(L')$, there exists such an element of L' that is not an element of $F(L)$. Without loss of generality, let $\{f'_q, f'_{q+1}, \cdots, f'_u\}$ be a set of such elements that are not included in $F(L)$. We prove the following three cases.

<u>Case-1 : $u = 1$.</u>

Since L' is $\{f'_u\}$, $F(L') = \{f'_u\}$. On the other hand, $f'_u \notin F(L)$. Thus $F(L) \cap F(L') = \phi$. Therefore, $|F(L) \cap F(L')| = 0 = 2^{u-1} - 1$.

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

(i) $F(L')$ is represented as follows:

$$F(L') = F(f'_1, f'_2, \cdots, f'_{u-1}) \cup \{f'_u\} \cup (f'_u \oplus F(f'_1, f'_2, \cdots, f'_{u-1})). \quad (5)$$

(ii) The following equation holds.

$$F(L) \cap (f'_u \oplus F(f'_1, f'_2, \cdots, f'_{u-1})) = \phi. \quad (6)$$

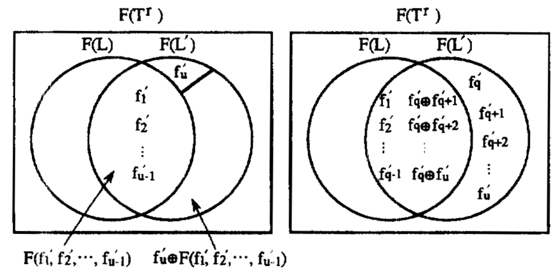
(iii) Since $f'_1, f'_2, \cdots, f'_{u-1}$ are elements of $F(L)$, $F(f'_1, f'_2, \cdots, f'_{u-1}) \subset F(L)$.

(iv) f'_u is not an element of $F(L)$

(v) From (i) $\sim$ (iv), the following equation holds.

$$F(L) \cap F(L') = F(f'_1, f'_2, \cdots, f'_{u-1}). \quad (7)$$

The set $\{f'_1, f'_2, \cdots, f'_{u-1}\}$ is $(u-1)$-independent. Thus, $|F(L) \cap F(L')| = 2^{u-1} - 1$



(a) q=u (b) $1 \leq q \leq u-1$

Figure 4 Relation between F(L) and F(L').

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

Since $f'_{q_1} \notin F(L)$ $(q \leq q_1 \leq u)$, f'_{q_1} is an element of $\{f_w\} \cup (f_w \oplus F(L))$, where f_w is such a linear function that the set $\{f_1, f_2, \cdots, f_{w-1}, f_w\}$ is w-independent. Thus, f'_{q_1} is represented as follows:

$$f'_{q_1} = f_w \oplus k_1^{q_1} f_1 \oplus k_2^{q_1} f_2 \oplus \cdots \oplus k_{w-1}^{q_1} f_{w-1}, \quad (8)$$

where there may exist the case that $(k_1^{q_1}, k_2^{q_1}, \cdots, k_{w-1}^{q_1}) = (0, 0, \cdots, 0)$. Thus, for $^\forall q_1$ $(q+1 \leq q_1 \leq u)$, the following equation holds.

$$f'_q \oplus f'_{q_1} = k_1^{q'_1} f_1 \oplus k_2^{q'_1} f_2 \oplus \cdots \oplus k_{w-1}^{q'_1} f_{w-1}. \quad (9)$$

$f'_q \oplus f'_{q_1}$ is therefore an element of $F(L)$.

On the other hand, let L'' be $\{ f'_1, f'_2, \cdots, f'_{q-1}, f'_q, f'_q \oplus f'_{q+1}, f'_q \oplus f'_{q+2}, \cdots, f'_q \oplus f'_u \}$, then L'' is u-independent, and subset of $F(L')$. Therefore, $F(L'') = F(L')$.

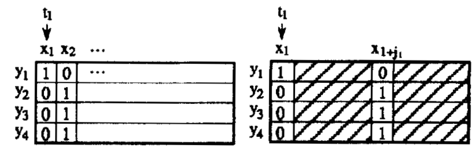
Therefore, relation between L'' and L is as same as the relation between L' and L in Case-2. Thus, $|F(L) \cap F(L'')| = 2^{u-1} - 1$. Consequently, $|F(L) \cap F(L')| = 2^{u-1} - 1$. ■

3.2 Proof that $|S^{w+j_1}| < |F(T^w)|$

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 $(a_2, a_3, a_4) = (1,0,0)$ 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 5(a)). Therefore, each of L_2^2, L_3^2 and L_4^2 is an empty set. Thus, $|S^2| = |F(L_2^2) \cup F(L_3^2) \cup F(L_4^2)| = 0$. ■

[Induction Step : $w = 1$] If $(a_2, a_3, a_4) = (1,0,0)$ or $(1,1,0)$, then the discussion in the basis step similarly holds. If $(a_2, a_3, a_4) = (1,1,1)$, then the general form of D_C becomes as shown in Figure 5(b). All elements of shadow area are 0s, since $w = 1$. But this is contradictory to the assumption that $X = X_1 \cup X_2 \cup \cdots \cup X_m$. In other words, if $j_1 > 1$, then there does not exist such a case that $(a_2, a_3, a_4) = (1,1,1)$. ■



(a) Basis Step (b) Induction Step

Figure 5 Dependence Matrix in case that w=1 and $(a_2, a_3, a_4) = (1, 1, 1)$.

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 $(1,1,0)$, 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 and c_4 are defined as shown in Figure 6(a), then $(c_i, b_i) \neq (1,1)$, since $w = 2$. Thus, $|L_2^3| \leq 1$, $|L_3^3| \leq 1$ and $|L_4^3| \leq 1$, and it is trivial that $L_{i_1}^3 = \phi$ for $^\exists i_1$ or $L_{i_1}^3 = L_{i_2}^3$ for $^\exists i_1$ and $^\exists i_2$ $(i_1 \neq i_2)$. Therefore, the following relation holds.

$$\begin{aligned} |S^3| &= |F(L_2^3) \cup F(L_3^3) \cup F(L_4^3)| \\ &= |F(L_{i_2}^3) \cup F(L_{i_3}^3)| \leq |F(L_{i_2}^3)| + |F(L_{i_3}^3)| \\ &\leq 2 < 3 = |F(T^2)|, \end{aligned} \quad (10)$$

where $i_3 \neq i_1$ and $i_3 \neq i_2$. ■

[Induction Step : $w = 2$] If $(a_2, a_3, a_4) = (1,0,0)$ or $(1,1,0)$, 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 (note that $|L_2^{2+j_1}| \leq 1$, $|L_3^{2+j_1}| \leq 1$ and $|L_4^{2+j_1}| \leq 1$, since $w = 2$):

If the second row vector does not have an 1-element in $(2 + j_1 - 1)$ successive columns starting with the first column, i.e., y_2 does not depend on each of inputs $x_1, x_2, \cdots, x_{2+j_1-1}$, then $L_2^{2+j_1} = \phi$. In this case, the following relation holds.

$$\begin{aligned} |S^{2+j_1}| &= |F(L_2^{2+j_1}) \cup F(L_3^{2+j_1}) \cup F(L_4^{2+j_1})| \\ &= |F(L_3^{2+j_1}) \cup F(L_4^{2+j_1})| \\ &\leq |F(L_3^{2+j_1})| + |F(L_4^{2+j_1})| \\ &\leq 2 < 3 = |F(T^2)|. \end{aligned} \quad (11)$$

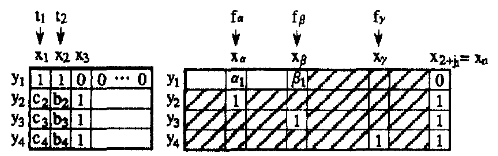
Similarly, we have $|S^{2+j_1}| < |F(T^2)|$ for the case that y_3 or y_4 does not depend on each of inputs $x_1, x_2, \cdots, x_{2+j_1-1}$. Thus, we assume that each of outputs y_2, y_3 and y_4 depends on one of $x_1, x_2, \cdots, x_{2+j_1-1}$ (this situation can occur only when $j_1 = n - w$, since $w = 2$). Let x_α, x_β and x_γ be such inputs for y_2, y_3 and y_4, respectively. If x_α and x_β are identical inputs, then the same relation as (11) holds. Similarly, we have $|S^{2+j_1}| < |F(T^2)|$ for the case that x_β and x_γ are identical inputs, or x_α and x_γ are identical inputs. Thus, in the discussions below, we assume that x_α, x_β and x_γ are different each other, and without loss of generality, we assume that $\alpha < \beta < \gamma$.

Figure 6(b) shows the general form of D_C under these assumptions, and α_1 and β_1 are defined as shown in the figure. 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_α, f_β and f_γ be linear functions which are assigned to x_α, x_β and x_γ, respectively. The first, second and third rows of the γth column vector are 0s. In the $(\gamma - w)$th visit of (A-3.1), i.e., in the assignment to x_γ, therefore, $S^\gamma = F(L_4^\gamma) = \phi$. Since t_1 is the smallest linear function of $F(T^2)$ ($\triangleq \{t_1, t_2, t_1 \oplus t_2\}$), therefore, t_1 is assigned to x_γ, i.e., $f_\gamma = t_1$.

(ii) If $\beta_1 = 1$, then $\alpha_1 = 1$ from Assumption-1 and $w = 2$, i.e., x_α and x_β are identical to x_1 and x_2, respectively. Thus, $f_\alpha = t_1$. From (i), therefore, $L_2^{2+j_1} = L_4^{2+j_1}$ in the assignment to x_{2+j_1}. Thus, the same relation as (11) holds.

(iii) If $\beta_1 = 0$, then the first, second and fourth rows of the βth column vector are 0s. Thus, $S^\beta = F(L_3^\beta) = \phi$. Therefore, $f_\beta = t_1$. From (i), therefore, $L_3^{2+j_1} = L_4^{2+j_1}$ in the assignment to x_{2+j_1}. Thus, we have $|S^{2+j_1}| < |F(T^2)|$ by replacing $L_3^{2+j_1}$ in (11) with $L_2^{2+j_1}$. ∎



(a) Basis Step (b) Induction Step
(x_α, x_β and x_γ are different each other)

Figure 6 Dependence Matrix in case that w=2 and $(a_2, a_3, a_4) = (1, 1, 1)$.

We assume that any D_C with the maximum row weight $(w-1)$ is $(w-1)$-assignable, and we prove that $|S^{w+j_1}| < |F(T^w)|$ for $^\forall j_1$ in any D_C with the maximum row weight w. The proof is done by induction with respect to j_1.

[Basis Step : $w \geq 3$] If $(a_2, a_3, a_4) = (1, 0, 0)$ or $(1, 1, 0)$, 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:

Let $q_i \triangleq |L_i^{w+1}|$ $(2 \leq i \leq 4;\ q_i \leq w - 1)$. If $F(L_{i_1}^{w+1}) \subseteq F(L_{i_2}^{w+1})$ for $^\exists i_1$ and $^\exists i_2$ $(i_1 \neq i_2)$, then the following relation holds.

$$\begin{aligned} |S^{w+1}| &= |F(L_2^{w+1}) \cup F(L_3^{w+1}) \cup F(L_4^{w+1})| \\ &= |F(L_{i_2}^{w+1}) \cup F(L_{i_3}^{w+1})| \\ &\leq 2^{q_{i_2}} - 1 + 2^{q_{i_3}} - 1 \leq 2^{w-1} - 1 + 2^{w-1} - 1 \\ &< 2^w - 1 = |F(T^w)|, \end{aligned} \quad (12)$$

where $i_3 \neq i_1$ and $i_3 \neq i_2$. If $q_{i_1} = 0$ $(2 \leq i_1 \leq 4)$, then the same relation as (12) holds. Thus, in the discussions below, we assume that $F(L_{i_1}^{w+1}) \not\subseteq F(L_{i_2}^{w+1})$ for $^\forall i_1$ and $^\forall i_2$ $(i_1 \neq i_2)$, and assume that $q_i \geq 1$ for $^\forall i$.

Without loss of generality, we assume that $w - 1 \geq q_2 \geq q_3 \geq q_4$, and prove the following four cases.

Case-1 : $w - 2 \geq q_2 \geq q_3 \geq q_4$

$$\begin{aligned} |S^{w+1}| &= |F(L_2^{w+1}) \cup F(L_3^{w+1}) \cup F(L_4^{w+1})| \\ &= |F(L_2^{w+1})| + |F(L_3^{w+1})| + |F(L_4^{w+1})| \\ &= 2^{q_2} - 1 + 2^{q_3} - 1 + 2^{q_4} - 1 \\ &= 2^{w-2} - 1 + 2^{w-2} - 1 + 2^{w-2} - 1 \\ &< 2^w - 1 = |F(T^w)|. \end{aligned} \quad (13)$$

Case-2 : $w - 1 = q_2,\ w - 2 \geq q_3 \geq q_4$

$$\begin{aligned} |S^{w+1}| &= 2^{q_2} - 1 + 2^{q_3} - 1 + 2^{q_4} - 1 \\ &= 2^{w-1} - 1 + 2^{w-2} - 1 + 2^{w-2} - 1 \\ &< 2^w - 1 = |F(T^w)|. \end{aligned} \quad (14)$$

Case-3 : $w - 1 = q_2 = q_3,\ w - 2 \geq q_4$

The following equation holds from Theorem 2.

$$\begin{aligned} |S^{w+1}| &= |F(L_2^{w+1}) \cup F(L_3^{w+1}) \cup F(L_4^{w+1})| \\ &= |F(L_2^{w+1})| + |F(L_3^{w+1})| + |F(L_4^{w+1})| \\ &\quad - |F(L_2^{w+1}) \cap F(L_3^{w+1})| \\ &\quad - |F(L_3^{w+1}) \cap F(L_4^{w+1})| \\ &\quad - |F(L_2^{w+1}) \cap F(L_4^{w+1})| \\ &\quad + |F(L_2^{w+1}) \cap F(L_3^{w+1}) \cap F(L_4^{w+1})| \\ &= 2^{q_2} - 1 + 2^{q_3} - 1 + 2^{q_4} - 1 \\ &\quad - (2^{q_3 - 1} - 1) - (2^{q_4 - 1} - 1) - (2^{q_4 - 1} - 1) \\ &\quad + |F(L_2^{w+1}) \cap F(L_3^{w+1}) \cap F(L_4^{w+1})| \\ &= 2^{q_2} + 2^{q_3 - 1} \\ &\quad + |F(L_2^{w+1}) \cap F(L_3^{w+1}) \cap F(L_4^{w+1})|. \end{aligned} \quad (15)$$

From Theorem 2, $|F(L_3^{w+1}) \cap F(L_4^{w+1})| = 2^{q_4 - 1} - 1$. On the other hand, $|F(L_2^{w+1}) \cap F(L_3^{w+1}) \cap F(L_4^{w+1})| \leq |F(L_3^{w+1}) \cap F(L_4^{w+1})|$. Therefore, the following relation holds.

$$|F(L_2^{w+1}) \cap F(L_3^{w+1}) \cap F(L_4^{w+1})| \leq 2^{q_4 - 1} - 1. \quad (16)$$

From (15) and (16), the following relation holds.

$$\begin{aligned} |S^{w+1}| &\leq 2^{q_2} + 2^{q_3 - 1} + 2^{q_4 - 1} - 1 \\ &\leq 2^{w-1} + 2^{w-2} + 2^{w-3} - 1 \\ &< 2^w - 1 = |F(T^w)|. \end{aligned} \quad (17)$$

Case-4 : $q_2 = q_3 = q_4 = w - 1$

Note that, for also this case, (15) holds. This case can occur if $x_{w+1} = x_n$ (see Figure 7(a)). If $b_i = 0$, then the jth column of the ith row vector is 1 $(2 \leq i \leq 4;\ 1 \leq j \leq w - 1)$, since $q_i = w - 1$. Thus, if $b_i = 0$, then $L_i^{w+1} = \{ t_1, t_2, \cdots, t_{w-1} \}$. Using this, the proof is done as follows:

Without loss of generality, we prove for the case that $(b_2, b_3, b_4) = (0, 0, 0)$ or $(1, 0, 0)$ or $(1, 1, 0)$.

If $(b_2, b_3, b_4) = (0, 0, 0)$ or $(1, 0, 0)$, then $L_3^{w+1} = L_4^{w+1} = \{ t_1, t_2, \cdots, t_{w-1} \}$. Consequently, $F(L_3^{w+1}) = F(L_4^{w+1})$. This is contradictory to the assumption that $F(L_3^{w+1}) \not\subseteq F(L_4^{w+1})$. In other words, if $F(L_3^{w+1}) \not\subseteq F(L_4^{w+1})$, then there does not exist such a case that $(b_2, b_3, b_4) = (0, 0, 0)$ or $(1, 0, 0)$.

If $(b_2, b_3, b_4) = (1, 1, 0)$, then both L_2^{w+1} and L_3^{w+1} contain t_w, and L_4^{w+1} never contains t_w. Thus, $F(L_2^{w+1}) \cap F(L_3^{w+1}) \supset F(L_2^{w+1}) \cap F(L_3^{w+1}) \cap F(L_4^{w+1})$. On the other hand, $|F(L_2^{w+1}) \cap F(L_3^{w+1})| = 2^{w-2} - 1$ from Theorem 2. Therefore, $|F(L_2^{w+1}) \cap F(L_3^{w+1}) \cap F(L_4^{w+1})| < 2^{w-2} - 1$. Thus, from (15), the following relation holds.

$$\begin{aligned} |S^{w+1}| &= 2^{q_2} + 2^{q_3 - 1} \\ &\quad + |F(L_2^{w+1}) \cap F(L_3^{w+1}) \cap F(L_4^{w+1})| \\ &< 2^{w-1} + 2^{w-2} + 2^{w-2} - 1 \end{aligned}$$

$$= \quad 2^w - 1 = |F(T^w)|. \qquad (18)$$

■

[Induction Step : $w \geq 3$] If $(a_2, a_3, a_4) = (1,0,0)$ or $(1,1,0)$, the proof is trivial from Theorem 1. And, if $(a_2, a_3, a_4) = (1,1,1)$, then the discussions until Case-3 in the basis step hold by replacing w+1 with w+j_1.

Thus, in the discussions below, we assume that, $q_2 = q_3 = q_4 = w - 1$, where $q_i \triangleq |F(L_i^{w+j_1})|$, and $F(L_2^{w+j_1}) \neq F(L_3^{w+j_1})$, $F(L_3^{w+j_1}) \neq F(L_4^{w+j_1})$ and $F(L_2^{w+j_1}) \neq F(L_4^{w+j_1})$ (see Figure 7(b)). Note that this case can occur when $j_1 = n - w$, i.e., $x_{w+j_1} = x_n$.

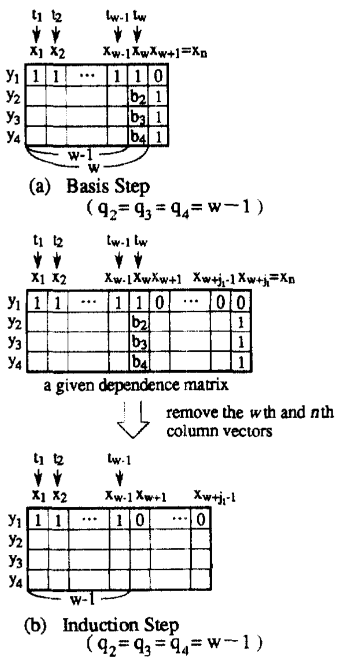


Figure 7 Dependence Matrix in case that $w \geq 3$ and $(a_2, a_3, a_4) = (1, 1, 1)$.

The maximum row weight of a new matrix constructed by removing the wth and nth column vectors from a given D_C is $w - 1$. On the other hand, we have assumed that any D_C with the maximum row weight $(w-1)$ is $(w-1)$-assignable. Thus, if the new matrix is considered as a new dependence matrix D'_C, the Akers' algorithm assigns a linear function constructed with some of $t_1, t_2, \cdots, t_{w-1}$ to each of inputs $x_1, x_2, \cdots, x_{w-1}, x_{w+1}, \cdots, x_{w+j_1-1}$. Therefore, for the original D_C, Akers' algorithm assigns a linear function constructed with some of $t_1, t_2, \cdots, t_{w-1}$ to each of inputs $x_1, x_2, \cdots, x_{w-1}, x_{w+1}, \cdots, x_{w+j_1-1}$, since the smallest linear function is assigned in the procedure (A-3.2.1). Thus, (i) if $b_i = 0$, then t_w never appears in the expression of any linear function of $L_i^{w+j_1}$. And from Assumption-3, t_w is assigned to x_w. Thus, (ii) if $b_i = 1$, then t_w is included in $L_i^{w+j_1}$. Using (i) and (ii), the proof is done as follows:

If $(b_2, b_3, b_4) = (0,0,0)$ or $(1,0,0)$, then t_w never appears in the expression of any linear function of $L_3^{w+j_1}$ and $L_4^{w+j_1}$. On the other hand, both $L_3^{w+j_1}$ and $L_4^{w+j_1}$ are $(w-1)$-independent. Therefore, $F(L_3^{w+j_1}) = F(L_4^{w+j_1})$. This is contradictory to the assumption that $F(L_3^{w+j_1}) \neq F(L_4^{w+j_1})$.

If $(b_2, b_3, b_4) = (1,1,0)$, then both $L_2^{w+j_1}$ and $L_3^{w+j_1}$ contain t_w, and consequently, both $F(L_2^{w+j_1})$ and $F(L_3^{w+j_1})$ contain t_w. On the other hand, t_w never appears in the expression of any linear function of $L_4^{w+j_1}$, and consequently, t_w never appears in the expression of any linear function of $F(L_4^{w+j_1})$. Therefore, $F(L_2^{w+j_1}) \cap F(L_3^{w+j_1}) \supset F(L_2^{w+j_1}) \cap F(L_3^{w+j_1}) \cap F(L_4^{w+j_1})$. Therefore, from Theorem 2, $2^{w-2} - 1 = |F(L_2^{w+j_1}) \cap F(L_3^{w+j_1})| > |F(L_2^{w+j_1}) \cap F(L_3^{w+j_1}) \cap F(L_4^{w+j_1})|$. On the other hand, it is trivial that an equation which is obtained by replacing $(w+1)$ in (15) with $(w+j_1)$ holds. Thus, the following relation holds.

$$\begin{aligned} |S^{w+j_1}| &= 2^{q_2} + 2^{q_3 - 1} \\ &\quad + |F(L_2^{w+j_1}) \cap F(L_3^{w+j_1}) \cap F(L_4^{w+j_1})| \\ &< 2^{w-1} + 2^{w-2} + 2^{w-2} - 1 \\ &= 2^w - 1 = |F(T^w)|. \qquad (19) \end{aligned}$$

■

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

[1] S. B. Akers, "On The Use of Linear Sums in Exhaustive Testing," *Digest of Papers 15th Symp. on Flt.Tolerant Comp.*, pp. 148-153, June 1985.

[2] H. Hiraishi, K. Kawahara and S. Yajima, "Locally Exhaustive Testing of Combinational Circuits Using Linear Logic Circuits," *Journal of Information Processing*, VOL. 11, No. 3, pp. 191-198, Nov. 1988.

[3] D. T. Tang and L. S. Woo, "Exhaustive Test Pattern Generation with Constant Weight Vector," *IEEE Trans. Comput.*, VOL. C-32, No. 12, pp. 1145-1150, Dec. 1983.

[4] L. T. Wang and E. J. McCluskey, "Linear Feedback Shift Register Design Using Cyclic Code," *IEEE Trans. Comput.*, VOL. C-37, No. 10, pp. 1302-1306, Oct. 1988.

[5] E. J. McCluskey, "Verification Testing — A Pseudoexhaustive Test Technique," *IEEE Trans. Comput.*, VOL. C-33, No. 6, pp. 541-546, June 1984.

[6] H. Michinishi, T. Yokohira and T. Okamoto, "Minimum Verification Test Set for Combinational Circuit," *Proc. ATS '92*, pp. 14-19, Nov. 1992.