*Engineering*

*Industrial & Management Engineering fields*

Okayama University *Year* 1992

# Minimum Verification test set for combinational circuit

Hiroyuki Michinishi
Okayama University

Tokumi Yokohira
Okayama University

Takuji Okamoto
Okayama University

# Minimum Verification Test Set for Combinational Circuit

Hiroyuki Michinishi, Tokumi Yokohira and Takuji Okamoto

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

## Abstract

*A sufficient condition under which a minimum verification test set (MVTS) for a combinational circuit has $2^w$ elements is derived, where $w$ is the maximum number of inputs on which any output depends, and an algorithm to find an MVTS with $2^w$ elements for any CUT with up to four outputs is described.*

## 1 Introduction

Many built-in self test (BIST) techniques have been proposed to reduce the cost of testing VLSI circuits [1,2,3]. A simple way to raise fault coverage in BIST is to test a combinational circuit under test (CUT) with $n$ inputs exhaustively, using $2^n$ test patterns. This way however raises a problem that too many test patterns are necessary if the CUT has many inputs.

In order to decrease the number of test patterns, retaining the advantages of exhaustive testing, McCluskey has proposed pseudoexhaustive test technique (verification test [4] ). If an output $f_i$ depends on $w_i$ inputs, a test set for the pseudoexhaustive test is generated so that $2^{w_i}$ patterns are applied to them. Hiraishi et al.[5] and Akers[6] have proposed algorithms to obtain a test set with $2^r$ test patterns ( $w \leq r \leq n$), where $w$ is the maximum number of inputs on which any output depends. The algorithms, however, do not guarantee to obtain a minimum test set. On the other hand, some researchers [7,8,9,10,11] have derived such a test set that can generate $2^w$ patterns applied to any $w$ inputs. The test set is therefore applicable to all the CUT with up to ${}_nC_w$ outputs. It is not however a minimum test set. In particular, if $w$ is large in a CUT with many inputs and few outputs, the number of test patterns is much larger than that of a minimum test set.

This paper investigates a minimum verification test set (MVTS) for the verification testing. In general, an MVTS has more than $2^w$ elements. We first derive a sufficient condition under which an MVTS has $2^w$ elements. This condition is derived from *the extended complete cyclic theorem* obtained by extending *the complete cyclic theorem*[12].

Next, we propose a method to find an MVTS for any CUT having an arbitrary number of inputs and up to four outputs. The first step is to create a matrix $RDP_C$ (*reduced partioned dependency matrix*) by the use of an algorithm obtained heuristically. The second step is to create an MVTS with $2^w$ elements by the use of the $RDP_C$.

## 2 Problem Statement

### 2.1 Definition of Minimum Verification Test Set (MVTS)

Figure 1 shows a combinational circuit under test (CUT) having $n$ inputs $x_n, x_{n-1}, \cdots, x_1$, and $m$ outputs $f_m, f_{m-1}, \cdots, f_1$. Let a set $S_{in}$ be $\{x_n, x_{n-1}, \cdots, x_1\}$, and let a set $I_i$ be $\{x^i_{n_i}, x^i_{n_i-1}, \cdots, x^i_1\}$ ($\subseteq S_{in}$) when $f_i$ depends on $x^i_{n_i}, x^i_{n_i-1}, \cdots, x^i_1$ ($1 \leq i \leq m$, and $|I_i| = n_i$). It is assumed that $I_m \cup I_{m-1} \cup \cdots \cup I_1 = S_{in}$ and the CUT remains combinational even if any fault occurs. *A verification test set*, VTS briefly, for the CUT is defined as follows[4].

**[Definition-I]** We call an $n$-dimensional vector ($x_n$, $x_{n-1}, \cdots, x_1$) *a test vector*. If a set $T$ of test vectors satisfies the following condition for $^\forall i$ ($1 \leq i \leq m$), then the set $T$ is a VTS.

*Condition:* The projection of $T$ onto ($x^i_{n_i}, x^i_{n_i-1}, \cdots, x^i_1$) subspace corresponding to $I_i$ contains all $2^{n_i}$ distinct binary patterns. □

Thus, the VTS is a set of test vectors which can exhaustively test each output of the CUT. If the number of test vectors of the VTS is minimal, then the VTS is an MVTS.

If $I_{i_1} \subseteq I_{i_2}$ for $^\exists i_1$, $^\exists i_2$ in a CUT, then we call that $f_{i_1}$ is *a covered output* by $f_{i_2}$. An output which is not a covered output is called *an essential output*. Let $T$ be a set of test vectors whose projection satisfies the condition in the definition of VTS for $I_i$ corresponding to each essential output. From the definition, it is trivial that the set $T$ is a VTS. Thus, in the discussions below, we assume that a CUT has only essential outputs.

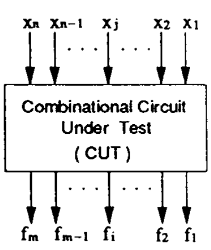



Figure 1 Structure of Combinational Circuit Under Test.

## 2.2 Redundancy of Universal Reduced Verification Test Set

McCluskey has already presented an approach to obtain a VTS[4]. This approach is divided into the following five steps.

The first step is to construct *a dependence matrix* $D_C$ for a CUT defined below.

**[Definition-II]** The dependence matrix $D_C$ for a CUT has $m$ row vectors and $n$ column vectors. Each row corresponds to one of the essential outputs of the CUT and each column corresponds to one of the inputs. An element is 1 iff the corresponding essential output depends on the corresponding input. All other elements are 0. □

Figure 2(a) shows an example of $D_C$.

The second step is to obtain *a partioned dependence matrix* ($DP_C$) defined below.

**[Definition-III]** The partioned dependence matrix $DP_C$ corresponding to a $D_C$ is formed by partioning the column vectors of the $D_C$ into sets such that: (i) each row vector of a set has at most one 1-entry, and (ii) the number of sets $p$ is a minimum. □

In this paper, each set mentioned above is referred to as *a partitioned set*. Figure 2(b) shows a $DP_C$ obtained from Figure 2(a). The $DP_C$ has six partitioned set ($p$ = 6), for example, the set of only the first column vector, and the set of the fourth and fifth column vectors.

The third step is to obtain *a reduced partioned dependency matrix* ($RDP_C$) defined below.

**[Definition-IV]** The reduced partitioned dependency matrix $RDP_C$ is obtained from $DP_C$ by merging all column vectors of each partitioned set into a single column vector in which a row has a 1 iff either of columns in the partioned set has a 1 in the corresponding row. Thus, $RDP_C$ has $m$ row vectors and $p$ column vectors. □

In this paper, the single column vector as mentioned above is referred to as *a merged column vector*. Figure 2(c) shows an $RDP_C$ obtained from Figure 2(b).

The fourth step is to obtain a VTS by regarding the $RDP_C$ as a new $D_C$. In the paper[4], such a VTS is called *a reduced verification test set* (RVTS).

The final step is to obtain a VTS for the original $D_C$ from an RVTS so that all columns of each partitioned set in the VTS are equal to the corresponding column in the RVTS.

Using these five steps, a VTS can be obtained. The second step is, however, a time consuming one due to the minimization of $p$ in the condition (ii), which is an NP-complete problem[4]. On the other hand, it is desirable in the fourth step to obtain a minimum RVTS (MRVTS) which has the minimum number of row vectors. It is however difficult to execute this step for $^\forall p$ and $^\forall w$ ($p \geq w$), where $w$ is the maximum row weight of the RVTS, i.e., $w \equiv max\{n_m, n_{m-1}, \cdots, n_1\}$. Thus, in the paper[4], assuming that a $DP_C$ can be obtained by some means, *a universal* RVTS (URVTS) defined below is derived as an RVTS in the fourth step.

**[Definition-V]** A matrix with $p$ column vectors is a universal reduced verification test set matrix $U(p, w)$ if all submatrices of $w$ columns of $U(p, w)$ contain all possible combination of $w$ binary digits. A universal RVTS is a set of all row vectors of a $U(p, w)$. □

For $p = w$ or $p = w + 1$, a URVTS becomes an MRVTS[4], and the number of elements of the MRVTS (MVTS) is $2^w$. If $p > w + 1$, it is not however guaranteed that a URVTS becomes an MRVTS, and the number of elements of a URVTS drastically increases as $p$ becomes larger.

Suppose that five steps mentioned above are executed under the condition that the second condition (ii) of the second step is omitted. In this case, a URVTS can be also obtained. It may have, however, more elements than a URVTS obtained with taking account of the condition (ii). Thus, it is desirable that $p$ is a minimum.

In this paper, we clarify that, if an $RDP_C$ obtained without taking account of the condition (ii) has a particular property described later, then an MRVTS (an MVTS) with $2^w$ elements can be constructed. In particular, we propose an algorithm to obtain an MVTS for any CUT having an arbitrary number of inputs $n$ and up to four essential outputs, and prove that the number of test vectors of the MVTS is $2^w$. In the succeeding discussions, unless otherwise stated, the terminologies $DP_C$ and $RDP_C$ means ones derived without considering the minimizing condition (ii).

| | $x_7$ | $x_6$ | $x_5$ | $x_4$ | $x_3$ | $x_2$ | $x_1$ |
|---|---|---|---|---|---|---|---|
| $f_1$ | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| $f_2$ | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| $f_3$ | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| $f_4$ | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

(a) Dc

| | $x_7$ | $x_6$ | $x_5$ | $x_4$ | $x_2$ | $x_3$ | $x_1$ |
|---|---|---|---|---|---|---|---|
| $f_1$ | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| $f_2$ | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| $f_3$ | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| $f_4$ | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

(b) DPc

| | $x_7$ | $x_6$ | $x_5$ | $x_4$ $x_2$ | $x_3$ | $x_1$ |
|---|---|---|---|---|---|---|
| $f_1$ | 1 | 0 | 0 | 1 | 1 | 1 |
| $f_2$ | 0 | 1 | 1 | 1 | 0 | 1 |
| $f_3$ | 0 | 1 | 1 | 1 | 1 | 0 |
| $f_4$ | 1 | 1 | 1 | 0 | 0 | 0 |

(c) RDPc

Figure 2 Steps to obtain a VTS

# 3 MVTS for a CUT with Covering Property

In this section, we show that, if an $RDP_C$ has a particular property described later, an MRVTS (an MVTS) with $2^w$ elements can be constructed. This is clarified by extending *the complete cyclic theorem* (CCT) described in our previous paper[12].

In the discussions below, the following notations are used.

- For an arbitrary matrix $M$, let $M[i, j]$ denote the $ij$ element of $M$.
- Let $M([i, j], p, q)$ denote a $p \times q$ submatrix of $M$ such that the uppermost and leftmost element of $M([i, j], p, q)$ corresponds to $M[i, j]$.
- When each of matrices $M_k, M_{k-1}, \cdots, M_1$ 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:

$$M \equiv M_k \bowtie M_{k-1} \bowtie \cdots \bowtie M_1.$$

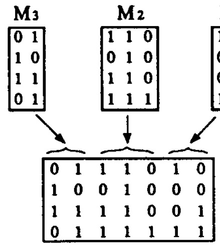
An example of the concatenation is shown in Figure 3.

## 3.1 Extended Complete Cyclic Theorem

*The extented complete cyclic theorem*, ECCT briefly, is given as follows (see Figure 4).

**[Extended Complete Cyclic Theorem (ECCT)]**

For $^\forall u (u \geq 2)$ and $^\forall v (v \geq 0)$, there exists a $2^u \times (u + v)$ matrix $M_v$ that satisfies the following conditions (C1), (C2) and (C3).




concatenated matrix $M_3 \bowtie M_2 \bowtie M_1$

Figure 3 Concatenation of matrices.

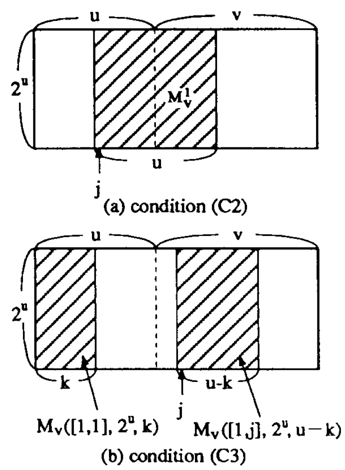



Figure 4 Extended Complete Cyclic Theorem.

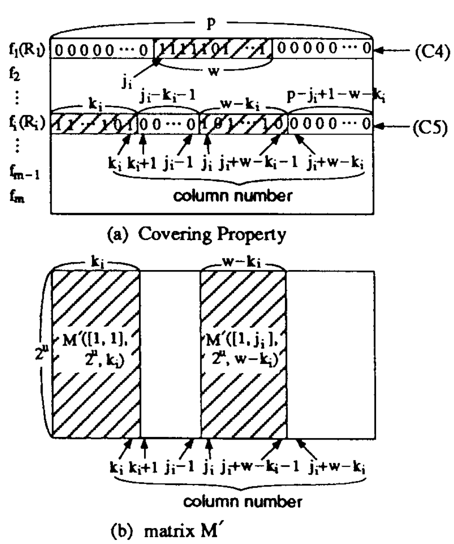



Figure 5 Covering Property.

(C1) The $i$th row vector of $M_v([1,1],2^u,u)$ is $i-1$ $(1 \leq i \leq 2^u)$, when it is regarded as a binary number.

(C2) For $^\forall j$ $(1 \leq j \leq v+1)$, if the submatrix $M_v^1$ is defined as follows,

$$M_v^1 \equiv M_v([1,j],2^u,u),$$

then, $M_v^1$ has $2^u$ distinct binary patterns (see Figure 4(a)).

(C3) For $^\forall k (1 \leq k \leq u-1)$ and $^\forall j (k+2 \leq j \leq v+k+1)$, if the submatrix $M_v^2$ of $M_v$ is defined as follows,

$$M_v^2 \equiv M_v([1,1],2^u,k) \bowtie M_v([1,j],2^u,u-k),$$

then, $M_v^2$ has $2^u$ distinct binary patterns (see Figure 4(b)). □

**[Proof]** It is trivial from (i) and (ii) below.

(i) If the value of $j$ in the condition (C3) is restricted to only $v+k+1$, the ECCT is identical to CCT which has been proved in our previous paper[12].

(ii) For $^\forall v_1$, $^\forall v$ $(v_1 \leq v)$, using the algorithm described in the paper[12], we can construct such a $2^u \times (u+v_1)$ matrix $M_{v_1}$ and a $2^u \times (u+v)$ matrix $M_v$ so that each of them satisfies the conditions (C1), (C2) and the restricted condition (C3), under the constraint that $M_{v_1} = M_v([1,1],2^u,u+v_1)$. □

## 3.2 Covering Property

Using the ECCT mentioned above, the following MRVTS theorem holds.

**[MRVTS theorem]** When an $RDP_C$ for a CUT has $m$ row vectors and $p$ column vectors, and its maximum row weight is equal to $w$, we represent the $RDP_C$ by $RDP_C(m,p,w)$

Assume that either of the following conditions (C4) and (C5) is satisfied in a matrix $M(\equiv RDP_C(m,p,w))$ for $^\forall i$ $(1 \leq i \leq m)$ (see Figure 5(a)).

(C4) There exists such $j_i$ $(1 \leq j_i \leq p-w+1)$ that all elements with 1 in the $i$th row vector $(= M([i,1],1,p))$ are included in the row vector $R_i$ $(\equiv M([i,j_i],1,w))$.

(C5) There exists such $k_i$ $(1 \leq k_i \leq w-1)$ and $j_i$ $(k_i+2 \leq j_i \leq p-w+k_i+1)$ that all elements with 1 in the $i$th row vector are included in the concatenated row vector $R_i$ $(\equiv M([i,1],1,k_i) \bowtie M([i,j_i],1,w-k_i))$.

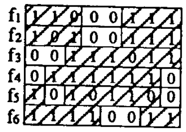
Then, a $2^w \times p$ matrix $M'$, which satisfies the conditions (C1), (C2) and (C3) by regarding $u=w$ and $v=p-w$, is an RVTS (an MRVTS) for the CUT. □

**[Proof]** The proof for case that (C5) is satisfied is as follows. $M''$ $(\equiv M'([1,1],2^u,k_i) \bowtie M'([1,j_i],2^u,w-k_i))$ has $2^w$ distinct binary patterns from ECCT (see Figure 5(b)). If we define the submatrix $M'''$ of $M''$ as follows,

$$M''' \equiv M''([1,z_{n_i}],2^w,1) \bowtie M''([1,z_{n_i-1}],2^w,1) \bowtie \cdots \bowtie M''([1,z_1],2^w,1),$$

where $R_i[1, z_q] = 1$ $(1 \le q \le n_i)$, then, $M'''$ has $2^{n_i}$ distinct binary patterns. The proof for case that (C4) can be performed in the similar way. □

When every row vector of an $RDP_C$ satisfies either (C4) or (C5), we call that the $RDP_C$ has *a covering property*. Figure 6 shows an example of $RDP_C$ with the covering property. The shadow area with $w$ elements in each row vector means that it covers every element whose value is 1. Thus, if an $RDP_C$ for a CUT has the covering property, then an MVTS with $2^w$ elements can be constructed using the following algorithm.




RDPc (6, 8, 6)

Figure 6 Example of RDPc with Covering Property.

**[MVTS algorithm I]**

(1) Create a matrix $M$ which satisfies three conditions in ECCT using the algorithm described in the paper[12], and regard this matrix $M$ as an MRVTS.

(2) Set all column vectors of each partitioned set in the MVTS with the corresponding column vector in the MRVTS. □

Note that $RDP_C$ $(m, p, p-1)$ and $RDP_C$ $(m, p, p)$ have the covering properties for $^\forall m$ and $^\forall p$. From MRVTS theorem, the numbers of elements of MRVTSs for $RDP_C(m, p, p-1)$ and $RDP_C(m, p, p)$ are $2^{p-1}$ and $2^p$, respectively.

## 4 MVTS for a CUT with up to Four Outputs

If $m = 1$, the MVTS has $2^n$ test vectors, and is easily obtained. Thus, we present the algorithms to obtain MRVTSs (MVTSs) in cases for $m = 2$, $m = 3$ and $m = 4$.

Principal idea of the algorithms is based on the fact that an $RDP_C$ has the covering property for $^\forall p$ and $^\forall w$. To show this, we introduce the following notations.

- Let $S(k_1, k_2, \cdots, k_m)$ denote a set of all $x_j$s such that the corresponding column vector in the dependence matrix $D_C$ is $(k_1, k_2, \cdots, k_m)^T$, where $v^T$ represents the transpose of vector $v$. For example, in the dependence matrix $D_C$ shown in Figure 2(a), $S(0,1,1,1) = \{x_6, x_5\}$, $S(1,0,1,0) = \{x_3\}$, $S(1,1,0,0) = \{x_2, x_1\}$.
- $S(k_1, k_2, \cdots, k_m)$ may be represented with $S_q$ briefly, where $q = \sum_{i=1}^{m} k_i \cdot 2^{m-i}$. Using this notation, the relation between $S_{in}$ and $S_q$ is represented as follows:
  - (i) For $^\forall q(1 \le q \le 2^m - 1)$, $S_q \subseteq S_{in}$.
  - (ii) $S_1 \oplus S_2 \oplus \cdots \oplus S_{2^m-1} = S_{in}$, i.e., $S_{in}$ can be partitioned into $S_1, S_2, \ldots, S_{2^m-1}$.
- Let $r_q$ be the number of elements of $S_q$, i.e., $r_q \equiv |S_q|$.
- For $^\forall x_j (\in S_{in})$, let $cv(x_j)$ be the corresponding column vector in a $D_C$.
- For $x_{j_k}, x_{j_{k-1}}, \cdots, x_{j_1}$ $(\in S_{in})$, and a column vector $(k_1, k_2, \cdots, k_m)^T$, if (i) a partitioned set can be constructed with $cv(x_{j_k}), cv(x_{j_{k-1}}), \cdots, cv(x_{j_1})$, and (ii) the merged column vector created from the partitioned set is equal to $(k_1, k_2, \cdots, k_m)^T$, then we call that the set $\{x_{j_k}, x_{j_{k-1}}, \cdots, x_{j_1}\}$ is *mergeable* with respect to $(k_1, k_2, \cdots, k_m)^T$. For example, in Figure 2(a) $\{x_7\}$ is mergeable with respect to $(1,0,0,1)^T$, and $\{x_4, x_2\}$ is mergeable with respect to $(1,1,1,0)^T$.

First, we describe an $RDP_C$ for a CUT with two essential outputs $(m = 2)$. Figure 7 shows an $RDP_C(2, n, w)$ $(p = n)$. Each arrow shows that the corresponding column vector in the $D_C$ is copied into the $RDP_C$ along it. The $RDP_C$ can be obtained using the following algorithm, which is called CA(2).

**[Algorithm CA(2)]**

(1) Construct the $D_C$ for the CUT.

(2) Set a temporary variable $j$ with 1, and execute the following procedure (2.1) and (2.2) for $q$ = 3, 2, 1 in this order until $S_q$ becomes empty.

  (2.1) Select an arbitrary element $x_{j_1}$ from $S_q$, and set the $j$th column vector of the $RDP_C$ with $cv(x_{j_1})$

  (2.2) Remove the element $x_{j_1}$ from $S_q$, and increase the value of $j$ by 1. □

Note that, in the procedure (2.1), if the binary representation of $q$ is $k_1k_2$, then $cv(x_{j_1})$ is equal to $(k_1, k_2)^T$.

In the algorithm CA(2), as seen in (2.1), a set of only the column vector $cv(x_{j_1})$ is a partitioned set. Thus, CA(2) creates an $RDP_C$ by rearranging column vectors of $D_C$ corresponding to the elements of $S_3, S_2, S_1$ in this order.

The followings (i) and (ii) are seen from the $RDP_C$ (2, $n$, $w$) shown in Figure 7.

(i) The shadow areas corresponding to $f_1$ and $f_2$ have $(r_3 + r_2)$ and $(r_3 + r_1)$ elements, respectively.

(ii) All elements of the shadow area corresponding to each of $f_1$ and $f_2$ are 1, and all elements of the other areas are 0.

From (i), (ii) and (iii) $w = max\{r_3 + r_2, r_3 + r_1\}$, the $RDP_C(2, n, w)$ has the covering property for $^\forall n$ and $^\forall w$.

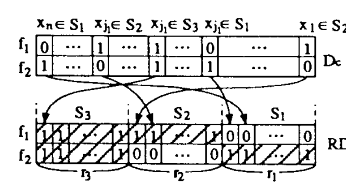



Figure 7 RDPc (2, n, w) for CUT with two essential outputs.

An $RDP_C$ for a CUT with three essential outputs ($m$ = 3) can be obtained with the following algorithm CA(3) (see Figure 8).

**[Algorithm CA(3)]** This is the same algorithm as CA(2) except that "$q$ = 3, 2, 1" in the procedure (2.2) is replaced with "$q$ = 7, 4, 6, 2, 3, 1, 5". □

The followings (i) and (ii) are seen from the $RDP_C$ (3, $n$, $w$) shown in Figure 8.

(i) The shadow areas corresponding to $f_1$, $f_2$ and $f_3$ have $(r_7+r_4+r_6+r_5)$, $(r_7+r_6+r_2+r_3)$ and $(r_7+r_3+r_1+r_5)$ elements, respectively.

(ii) All elements of the shadow area corresponding to each of $f_1$, $f_2$ and $f_3$ are 1, and all elements of the other areas are 0.

From (i), (ii) and (iii) $w = max\{r_7+r_4+r_6+r_5, r_7+r_6+r_2+r_3, r_7+r_3+r_1+r_5\}$, the $RDP_C(3, n, w)$ has the covering property.

As mentioned above, the $RDP_C$ for $m$ = 2 and $m$ = 3 can be generated by simply rearranging the column vectors of the $D_C$ so that it has the covering property.

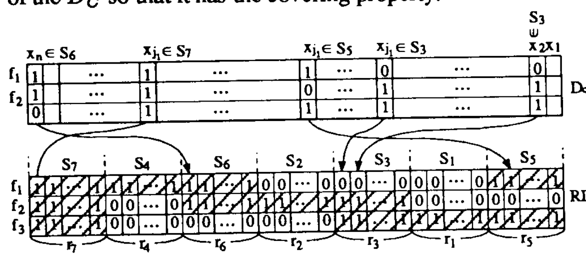



Figure 8 RDPc (3, n, w) for CUT with three essential outputs.

Next, prior to the algorithm for $m$ = 4, we describe how to merge column vectors in $D_C$, which correspond to elements of some $S_q$s. For example, from the definition of $S(k_1, k_2, \cdots, k_m)$, for $^\forall x_{j_1}(\in S_6)$, $x_{j_2}(\in S_9)$, $\{x_{j_1}, x_{j_2}\}$ is mergeable with respect to $(1, 1, 1, 1)^T$, which can be regarded as if it is an element of $S_{15}$. In the same way, $\{x_{j_1}, x_{j_2}\}$ $(x_{j_1} \in S_8, x_{j_2} \in S_3)$ is mergeable with respect to $(1, 0, 1, 1)^T$, and $\{x_{j_1}, x_{j_2}, x_{j_3}\}$ $(x_{j_1} \in S_1, x_{j_2} \in S_2, x_{j_3} \in S_4)$ is mergeable with respect to $(0, 1, 1, 1)^T$.

The algorithm for $m$ = 4 is constructed by the merging principle mentioned above, but somewhat complicated, because the number of $S(k_1, k_2, k_3, k_4)$s increases up to 15. So, we separate into 8 cases by combinations of signs in three differences $|S_{12}| - |S_3|$, $|S_{10}| - |S_5|$ and $|S_9| - |S_6|$. Here, we show for the case that all signs are non-negative (see Figure 9).

**[Algorithm CA(4)]**

(1) Construct the $D_C$ for the CUT.

(2) Set a temporary variable $j$ with 1, and execute following procedures (2.1), (2.2) and (2.3) for $q$ = 3, 5, 6 in this order until $S_q$ becomes empty.

(2.1) Select two arbitrary elements $x_{j_1}$ and $x_{j_2}$ from $S_q$ and $S_{15-q}$, respectively, and create a partitioned set from $cv(x_{j_1})$ and $cv(x_{j_2})$.

(2.2) Create a merged column vector from the partition set obtained in (2.1), and set the $j$th column vector of the $RDP_C$ with the merged column vector.

(2.3) Remove the elements $x_{j_1}$ and $x_{j_2}$ from $S_q$ and $S_{15-q}$, respectively, and increase the value of $j$ by 1.

(3) Execute the following procedure (3.1) for $q$ = 15, 9, 11, 10, 14, 7, 13, 12 in this order.

(3.1) Let $k_1k_2k_3k_4$ be the binary representation of $q$, and then repeat the following procedures (3.1.1), (3.1.2) and (3.1.3) if there exists such $x_{j_k}$ $(\in S_{q_k})$, $x_{j_{k-1}}$ $(\in S_{q_{k-1}})$, $\cdots$, $x_{j_1}$ $(\in S_{q_1})$ that the set $\{x_{j_k}, x_{j_{k-1}}, \cdots, x_{j_1}\}$ is mergeable with respect to $(k_1, k_2, k_3, k_4)^T$.

(3.1.1) Create a partitioned set from $cv(x_{j_k})$, $cv(x_{j_{k-1}})$, $\cdots$, $cv(x_{j_1})$.

(3.1.2) Create a merged column vector from the partition set obtained in (3.1.1), and set the $j$th column vector of the $RDP_C$ with the merged column vector.

(3.1.3) Remove the elements $x_{j_k}$, $x_{j_{k-1}}$, $\cdots$, $x_{j_1}$ from $S_{q_k}$ $S_{q_{k-1}}$ $\cdots$ $S_{q_1}$, respectively, and increase the value of $j$ by 1.

(4) Execute the following procedures (4.1), (4.2) and (4.3) for $q$ = 8, 4, 2, 1 until $S_q$ becomes empty.

(4.1) Select an arbitrary element $x_{j_1}$ from $S_q$, and set the $j$th column vector of the $RDP_C$ with $cv(x_{j_1})$

(4.2) Remove the element $x_{j_1}$ from $S_q$, and increase the value of $j$ by 1. □

Note that, (i) each of the merged column vectors created in the procedure (2.2) is equal to $(1, 1, 1, 1)^T$, (ii) each of the merged column vectors created in the procedure (3.1.2) is equal to $(k_1, k_2, k_3, k_4)^T$, and (iii) at most only one $S_q(q$ = 8, 4, 2, 1) is not empty before the procedure (4) is executed.

In Figure 9, if the binary representation of $q$ is $k_1k_2k_3k_4$, $T_q$ represents a group of sets of inputs which are mergeable with respect to $(k_1, k_2, k_3, k_4)^T$. All the elements in the shadow area are 1, and the others are 0. The number of ele-

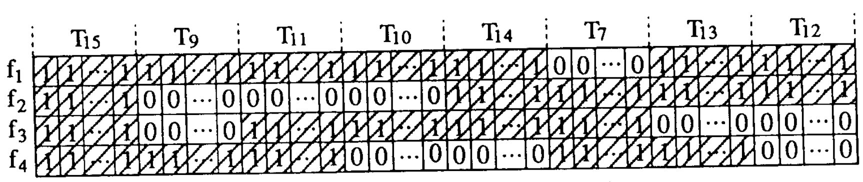



Figure 9 RDPc (4, n, w) for CUT with four essential outputs.

ments of the shadow areas corresponding to $f_i$ is equal to $n_i$ ($1 \leq i \leq 4$). The $RDP_C$ has therefore the covering property, since $w \equiv max\{n_1, n_2, n_3, n_4\}$. We can also show similarly that there exists an $RDP_C$ with the covering property in any other combinations of signs.

For $m$ ($2 \leq m \leq 4$), an $RDP_C$ with the covering property, as mentioned above, can be constructed. From the MRVTS theorem described in 3.2, there exists an MRVTS (MVTS) with $2^w$ elements. An MVTS can easily derived using the following algorithm.

**[MVTS algorithm II]**

(1) Using CA($m$) ($2 \leq m \leq 4$), construct an $RDP_C$ with the covering property.

(2) Execute MVTS algorithm I described in 3.2. □

Using MVTS algorithm II-(1), an $RDP_C$( Figure 10(a)) is obtained from Figure 2(a). From MVTS algorithm II-(2), an MRVTS(Figure 10(b)) and the corresponding MVTS (Figure 10(c)) are derived. The MVTS has $2^4 (w = 4)$ elements, while a VTS which is obtained from the paper[4] has 21 elements.

| | $x_7$ | $x_3$ | $x_2$ ($x_4$) | $x_6$ | $x_5$ | $x_1$ |
|---|---|---|---|---|---|---|
| $f_1$ | 1 | 1 | 1 | 0 | 0 | 1 |
| $f_2$ | 0 | 0 | 1 | 1 | 1 | 1 |
| $f_3$ | 0 | 1 | 1 | 1 | 1 | 0 |
| $f_4$ | 1 | 0 | 0 | 1 | 1 | 0 |

(a) RDPc

| $x_7$ | $x_3$ | $x_2$ ($x_4$) | $x_6$ | $x_5$ | $x_1$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 |

(b) MRVTS

| $x_7$ | $x_6$ | $x_5$ | $x_4$ | $x_3$ | $x_2$ | $x_1$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 |

(c) MVTS

Figure 10 RDPc, MRVTS and MVTS obtained using MVTS algorithm II.

## 5 Conclusion

We first derived a sufficient condition under which an MVTS for a CUT has $2^w$ elements. Second, we proposed an algorithm to find an MTVS for any CUT with up to four essential outputs, and proved that the MVTS has $2^w$ elements independent of the number of the inputs.

It is an open problem whether every CUT with five outputs has an MVTS with $2^w$ elements or not. It has however been shown[4] that some CUT with six outputs has an MVTS with more than $2^w$ elements.

## References


[1] T. W. Williams and K. P. Parker, "Design for Testability — A survey," *Proc. IEEE*, VOL. 71, pp. 98-112, Jan. 1983.

[2] T. W. Williams, "VLSI Testing," *Computer*, VOL. 17, pp. 126-136, Oct. 1984.

[3] E. J. McCluskey, "Built-In Self-Test Techniques," *IEEE Design and Test Comput.*, VOL. 2, pp. 21-28, April 1985.

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

[5] 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.

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

[7] Z. Barzilai, D. Coppersmith and A. L. Rosenberg, "Exhaustive Generation of Bit Patterns with Applications to VLSI Self-Testing," *IEEE Trans. Comput.*, VOL. C-32, No. 2, pp. 190-194, Feb. 1983.

[8] 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.

[9] D. T. Tang and C. L. Chen, "Logic Test Pattern Generation Using Linear Codes," *IEEE Trans. Comput.*, VOL. C-33, No. 9, pp. 845-850, Sept. 1984.

[10] C. L. Chen, "Exhaustive Test Pattern Generation Using Cyclic Codes," *IEEE Trans. Comput.*, VOL. C-37, No. 2, pp. 225-228, Feb. 1988.

[11] 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.

[12] T. Yokohira, H. Michinishi, T. Okamoto and Y. Sugiyama, "Test Set for a Multibit Shifter Constructed with Multiplexers," *Trans. IEICE*, VOL. E 73, No. 8, pp. 1301-1309, Aug. 1990.