

# Locally Exhaustive Testing of Combinational Circuits Using Linear Logic Circuits

HIROMI HIRAISHI\*, KENJI KAWAHARA\*\* and SHUZO YAJIMA\*

Locally exhaustive testing of multiple output combinational circuits is the test which provides exhaustive patterns for each set of inputs on which each output is truly dependent. In this paper, a method for locally exhaustive testing of combinational circuits using linear logic circuits is proposed. The testing scheme consists of a linear combinational circuit and a linear feedback shift register. This scheme can be implemented with smaller amounts of hardware than other existing methods, and can be applied to built-in self testing. The correspondence problem between the outputs of the linear combinational circuit and the inputs of the circuit under test is formalized as a vector assignment problem of hypergraphs. It resembles the coloring problem of graphs and is also proved to be NP-complete. The vector assignment of perfect hypergraphs is shown to be equivalent to the construction of linear codes. Several graph-theoretical properties which can be utilized to reduce the size of hypergraphs are shown. Finally, the length of test sequences is discussed.

## 1. Introduction

Logic circuit testing plays an important part in the improvement of the reliability of logic circuits. The rapid progress of semiconductor technology allows us to design large-scale logic circuits, but as a consequence it makes logic testing a very difficult process. This paper is concerned with one of the approaches to ease the difficulties of logic testing: locally exhaustive testing of combinational circuits.

In logic circuit testing, a circuit is tested by providing a test sequence and examining its corresponding responses. The test sequence is usually designed so that it can detect all probable faults in the circuit. But as the number of gates in the circuit increases, it becomes very difficult and time consuming to generate test sequences. It is proved to be NP-complete to obtain a test pattern which detects a specified fault in combinational circuits [1][2]. In addition, we face the problem of fault models. Although the single stuck-at fault assumption has been widely adopted, until now, it seems inappropriate for highly integrated circuits.

In order to overcome the above problems, exhaustive testing where the circuits are exercised with all possible input patterns can be considered. In exhaustive testing, test sequences need not be calculated and the functions of the circuits can be uniquely determined, whereas test sequences are impractically long for the circuits with large number of inputs.

There are, however, many multiple output circuits

where each output depends on only a small portion of the inputs. In order to test such circuits, not all possible input patterns are necessarily applied but it is sufficient to apply a sequence that provides exhaustive patterns for each set of inputs on which each output depends. This testing method is called "locally exhaustive testing of combinational circuits" in this paper. In locally exhaustive testing, test sequences are shortened without spoiling the testing capability of exhaustive testing.

Concerned with locally exhaustive testing of combinational circuits, researches have been made mainly on the locally exhaustive test sequences for all the subsets of inputs of size  $r$  where  $r$  is the maximum number of the inputs on which an output is dependent [3]-[7][15]. Though their test sequences can be used in locally exhaustive testing, it seems impractical to use these sequences directly for actual circuits, because they often become quite long. Different from these methods, McCluskey proposes a locally exhaustive testing in which test sequences are calculated considering the input-output dependencies of the circuit to be tested [8].

In this paper, a testing scheme using linear logic circuits is newly proposed as a test sequence generator for locally exhaustive testing of combinational circuits [16][17]. In our scheme, test sequence generator is composed of a linear feedback shift register and a linear combinational circuit and can be designed much more compactly than those in other methods. It is next shown that the correspondence problem between the inputs of the circuit under test and the outputs of the linear combinational circuit can be formalized as a vector assignment problem of hypergraphs. This vector assignment problem of hypergraphs is proved to be NP-complete, and its relation to linear code is argued. The bounds for

\*Department of Information Science, Faculty of Engineering, Kyoto University.

\*\*He is now at Computer Systems Laboratories, Sharp Corporation.

$$D = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{matrix} \\ \begin{matrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{matrix} & \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \end{matrix}$$

Fig. 1 A dependence matrix  $D$ .

vector assignment numbers and the graph-theoretical properties on this problem are also discussed. Finally the performance of locally exhaustive testing using linear logic circuits is discussed.

## 2. Preliminaries

We shall consider a 2-value,  $n$ -input,  $m$ -output combinational logic circuit  $C$ . The set of input lines of  $C$  and the set of its output lines are named  $X = \{x_1, x_2, \dots, x_n\}$  and  $Y = \{y_1, y_2, \dots, y_m\}$ , respectively. In general an output  $y_i (\in Y)$  depends only on a subset of  $X$ . The dependence matrix  $D = \{d_{ij}\}$  of  $C$  is defined as an  $m$  by  $n$  matrix whose element  $d_{ij}$  is equal to 1 if the output  $y_i$  depends on the input  $x_j$ , and is equal to 0 otherwise. Fig. 1 shows an example of a dependence matrix. The drive input set  $DR_i$  of an output  $y_i$  is defined as the set of inputs on which  $y_i$  is dependent, i.e.,  $DR_i = \{x_j | 1 \leq j \leq n, d_{ij} = 1\}$ .

Combinational circuits are usually tested by feeding a set of test patterns and examining the corresponding outputs. A binary  $n$ -dimensional vector  $(t_1, t_2, \dots, t_n)$  is regarded as a test pattern for  $C$ . Each input  $x_i (\in X)$  is provided with a logical value  $t_i$  by the test pattern and outputs for it will be examined. A  $t$  by  $n$  matrix  $T$  whose row vectors are test patterns for  $C$  is called a test sequence for  $C$ , where  $t$  is the number of test patterns and is called the length of test sequence  $T$ .

The set of all binary  $k$ -dimensional vectors is denoted by  $V^k$ . Vectors  $v_1, v_2, \dots, v_p$  contained in  $V^k$  are said to be linearly independent if and only if there exist no sets of coefficients  $\{a_1, a_2, \dots, a_p\}$  ( $a_i \in \{0, 1\}$ ) such that

at least one coefficients  $a_i \neq 0$ , and

$$a_1 \cdot v_1 + a_2 \cdot v_2 + \dots + a_p \cdot v_p = (0, 0, \dots, 0),$$

where  $\cdot$  and  $+$  mean the scalar product and the bitwise addition over modulo 2, respectively.

The pair  $H = (X, E)$  is called a hypergraph [9], where  $X = \{x_1, x_2, \dots, x_n\}$  is a finite set and  $E = \{e_1, e_2, \dots, e_m\}$  is a set of subsets of  $X$  that satisfies  $e_i \neq \phi$  and  $\bigcup_{i=1}^m e_i = X$ . Each element  $x_j$  in  $X$  is called a vertex and each set  $e_i$  in  $E$  is called an edge. The incidence matrix of a hypergraph  $H = (X, E)$  is denoted by  $D = \{d_{ij}\}$  and is an  $m$  by  $n$  matrix whose element  $d_{ij}$  is equal to 1 if  $x_j \in e_i$  and is equal to 0 if  $x_j \notin e_i$ . A corresponding hypergraph is one obtained by regarding a dependence matrix as its incidence matrix. Fig. 2 shows the corresponding

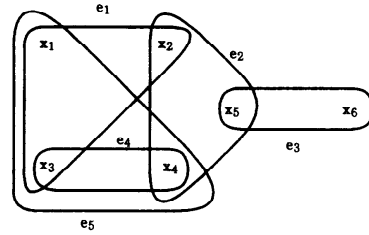
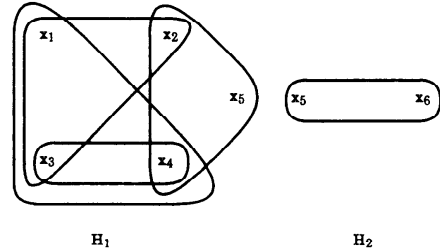
Fig. 2 A corresponding hypergraph  $H$ .

Fig. 3 Separation of a hypergraph into pieces by articulation set.

hypergraph  $H$  of the dependence matrix  $D$  in Fig. 1.

$|e_i|$  is called the rank of  $e_i$  where  $|S|$  represents the number of elements in a set  $S$ . A hypergraph, all edges of which have rank two, is called a graph. The rank of hypergraph  $H$  is the maximum rank of the edges in  $H$ . Perfect hypergraph of rank  $r$  is the hypergraph  $H = (X, E)$ , where  $E$  consists of all the subsets of size  $r$  of  $X$  and  $H_{n,r}$  denotes the perfect hypergraph of rank  $r$  with  $n$  vertices. The subhypergraph of  $H = (X, E)$  generated by the set  $A \subseteq X$  is defined to be the hypergraph  $H_A = (A, E_A)$ , where  $E_A = \{e_i \cap A | e_i \in E, e_i \cap A \neq \phi\}$ .

An edge  $e_i$  of hypergraph  $H = (X, E)$  is said to be redundant if there exists an edge  $e_j$  such that 1)  $e_i \subset e_j$  or 2)  $e_i = e_j$  and  $i > j$ . A hypergraph with no redundant edges is said to be irreducible. A reduced form of hypergraph  $H$  is the irreducible hypergraph obtained by omitting all redundant edge from  $H$ .

Connected components of a hypergraph and connected hypergraphs are defined in the same way as in usual graphs. If there exists two edges  $e_{a_1}, e_{a_2}$  in a connected hypergraph  $H = (X, E)$  such that

1)  $Q = e_{a_1} \cap e_{a_2}$  and

2) the subhypergraph  $H_{X-Q}$  generated by the set  $X-Q$  has  $s (\geq 2)$  connected components  $H_i = (X_i, E_i)$  ( $i = 1, 2, \dots, s$ ), the pair of  $e_{a_1}$  and  $e_{a_2}$  is called the articulation pair and  $Q$  is called the articulation set. Pieces by  $Q$  are defined as the subhypergraphs  $H_{X_i \cup Q}$  generated by the set  $X_i \cup Q$ . The hypergraph  $H$  shown in Fig. 2 has an articulation set  $Q = \{x_3\}$ , and its pieces by  $Q$  is shown in Fig. 3.

$$T = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

Fig. 4 Locally exhaustive test sequence.

### 3. Locally Exhaustive Testing Using Linear Logic Circuits

#### 3.1 Concept of Locally Exhaustive Testing

**Definition 1.** A test sequence  $T$  is said to be a locally exhaustive test sequence for a subset of inputs  $X' = \{x_{i_1}, x_{i_2}, \dots, x_{i_p}\} (\subseteq X)$  of a combinational circuit  $C$  if all the vectors in  $V^p$  are contained as row vectors in the submatrix  $T'$  which consists of the  $x_{i_1}$ -th,  $x_{i_2}$ -th,  $\dots$ ,  $x_{i_p}$ -th column vectors of  $T$ .

**Definition 2.** A test sequence  $T$  is said to be a locally exhaustive test sequence for a combinational circuit  $C$  if  $T$  is a locally exhaustive test sequence for all the drive input sets of  $C$ .

**Example 1.** The test sequence  $T$  illustrated in Fig. 4 is a locally exhaustive test sequence for  $\{x_1, x_2\}$ , but not for  $\{x_2, x_3\}$ .

As for locally exhaustive testing of combinational circuits, research has concentrated on the locally exhaustive test sequences for all the subsets of inputs of size  $r$ , where  $r$  is the maximum number of elements in the drive input sets. Although these test sequences can be regarded as solutions of locally exhaustive test sequences, test sequences for actual circuits need not necessarily be such a locally exhaustive test sequence.

In [3][6] shorter test sequences with such testing capabilities are offered by constructive methods, but it costs very much to generate test sequences by these methods. In [4][5] locally exhaustive test sequences which can be systematically generated are proposed, but their test sequences often become impractically long. Different from these methods, McCluskey proposes a locally exhaustive testing in which test sequences are calculated using the input-output dependence information of the circuits under test [8].

The testing method proposed in this paper also uses the input-output dependencies, but specific linear logic circuits are newly proposed and its testing schemes are generally more compact than those in [8].

#### 3.2 Utilization of Linear Logic Circuits

A linear feedback shift register (LFSR) is a circuit composed of a shift register and a feedback loop with modulo 2 adders. For a LFSR with  $k$  stages, the feedback is generally characterized by a  $k$ -degree polynomial over  $GF(2)$ , the characteristic polynomial. If a primitive polynomial [10] of degree  $k$  is adopted as the characteristics polynomial of LFSR, all the vectors

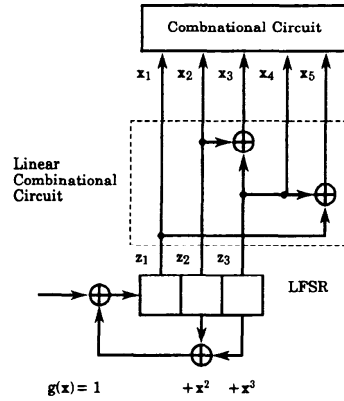


Fig. 5 Locally exhaustive testing using linear logic circuits.

in  $V^k$  are generated at the outputs.

Consider the scheme composed of a LFSR and a linear combinational circuit (LCC) as shown in Fig. 5. The LFSR has outputs  $z_1, z_2, \dots, z_k$  and all the patterns in  $V^k$  are observed at outputs  $z_1, z_2, \dots, z_k$  as time goes on (In Fig. 5,  $k=3$ ). The LCC offers some linear combinations of the outputs of the LFSR. Its outputs are in the form of  $a_1 \cdot z_1 \oplus a_2 \cdot z_2 \oplus \dots \oplus a_k \cdot z_k$ , where  $a_1, a_2, \dots, a_k \in \{0, 1\}$ ,  $\cdot$  and  $\oplus$  mean logical AND and modulo 2 addition, respectively.

An input  $x_i$  of the circuit under test is said to correspond to a binary  $k$ -dimensional vector  $v_i = (a_1, a_2, \dots, a_k)$ , when  $x_i$  is connected to the output  $a_1 \cdot z_1 \oplus a_2 \cdot z_2 \oplus \dots \oplus a_k \cdot z_k$  of the LCC. The sequence which is observed at inputs  $x_1, x_2, \dots, x_n$  of the circuit under test while all the vectors in  $V^k$  are generated at outputs  $z_1, z_2, \dots, z_k$  is denoted by  $T_{all}$ .

**Theorem 1.**  $T_{all}$  is a locally exhaustive test sequence for a set of inputs  $\{x_{i_1}, x_{i_2}, \dots, x_{i_p}\}$  if and only if the associated vectors  $v_{i_1}, v_{i_2}, \dots, v_{i_p}$  are linearly independent.

**Proof:** Let  $f$  denote the linear transformation represented by the matrix whose row vectors are  $v_{i_1}, v_{i_2}, \dots, v_{i_p}$ .

When  $v_{i_1}, v_{i_2}, \dots, v_{i_p}$  are linearly independent, the rank of  $f$  is  $p$  and image of  $V^k$  by  $f$  is  $V^p$ . Therefore the set of inputs  $\{x_{i_1}, x_{i_2}, \dots, x_{i_p}\}$  is provided with all vectors in  $V^p$  and  $T_{all}$  is a locally exhaustive test sequence for the set of inputs.

When  $v_{i_1}, v_{i_2}, \dots, v_{i_p}$  are not linearly independent, on the other hand, the rank of  $f$  is  $p'$  which is less than  $p$  and the image of  $V^k$  by  $f$  is a proper subset of  $V^p$ . Therefore the set of inputs  $\{x_{i_1}, x_{i_2}, \dots, x_{i_p}\}$  is not provided with all vectors in  $V^p$  and  $T_{all}$  is not a locally exhaustive test sequence for the set of inputs. QED

**Example 2.** In the testing scheme shown in Fig. 5, the vectors corresponding to each input of the circuit under test are as follows:  $v_1 = (100)$ ,  $v_2 = (010)$ ,  $v_3 = (011)$ ,  $v_4 = (001)$ ,  $v_5 = (101)$ . As  $v_1, v_2$  and  $v_3$  are linearly independent, the test sequence  $T_{all}$  defined above is a local-

ly exhaustive test sequence for the set of inputs  $\{x_1, x_2, x_3\}$ . On the other hand, as  $v_1, v_4$  and  $v_5$  are not linearly independent (for  $v_1 + v_4 = v_5$ ),  $T_{\text{all}}$  is not a locally exhaustive test sequence for the set of inputs  $\{x_1, x_4, x_5\}$ .

When the dependence matrix of the circuit under test is given, the testing scheme for locally exhaustive testing of the circuit is obtained by corresponding the outputs of the LCC to the inputs of the circuit so that the condition of linear independence stated in Theorem 1 is satisfied for every drive input set.

Although the result of Theorem 1 is true so long as exhaustive patterns appear at the inputs of LCC, it is recommended to use LFSRs because LFSRs can be designed compactly and generate exhaustive patterns quickly. Therefore, the testing scheme composed of an LFSR and an LCC is good for the test sequence generator for locally exhaustive testing of combinational circuits. This testing method is hereafter called locally exhaustive testing using linear logic circuits.

#### 4. Vector Assignment Problem of Hypergraphs

In the preparation of locally exhaustive testing, the correspondence problem between the inputs of the circuit under test and the outputs of the LCC has to be solved. This correspondence problem is formalized as a vector assignment problem of a corresponding hypergraph obtained by regarding the dependence matrix as its incidence matrix.

**Definition 3.** A hypergraph  $H=(X, E)$  is said to be  $k$ -vector assignable if there exists an assignment of the binary  $k$ -dimensional vectors in  $V^k$  to the vertices of  $H$  such that for every edge  $e_i$  the assigned vectors  $v_{i_1}, v_{i_2}, \dots, v_{i_p}$  to the vertices  $x_{i_1}, x_{i_2}, \dots, x_{i_p}$  in  $e_i$  are linearly independent.

**Definition 4.** The vector assignment problem of hypergraphs is the following decision problem.

Instance: Hypergraph  $H=(X, E)$  and a positive integer  $k$ .

Question: Is  $H$   $k$ -vector assignable?

**Definition 5.** The vector assignment number of a hypergraph  $H$  is defined as the smallest number of  $k$  such that  $H$  is  $k$ -vector assignable and is denoted by  $\delta(H)$ .

Obviously,  $\delta(H)$  has the following properties.

**Lemma 1.** For a hypergraph  $H=(X, E)$  whose rank is  $r$ ,  $r \leq \delta(H) \leq |X|$ .

##### 4.1 Proof of NP-Completeness

The vector assignment problem of hypergraphs is proved to be an NP-complete problem by showing that the coloring problem of graphs is polynomially transformable to it.

**Definition 6.** A graph  $G=(X, E)$  is said to be  $k$ -colorable if there exists an assignment of  $k$  colors to the vertices of  $G$  such that no two vertices contained in a common edge are assigned the same colors.

**Definition 7.** The chromatic number of a graph  $G$  is

defined as the smallest number of  $k$  such that  $G$  is  $k$ -colorable and is denoted by  $\chi(G)$ .

The coloring problem of graphs (Is a graph  $G$   $k$ -colorable?) is well known to be NP-complete [11]. The vector assignment problem of hypergraphs resembles the coloring problem of graphs.

**Theorem 2.** The vector assignment problem of hypergraphs is NP-complete.

**Proof:** Without loss of generality, hypergraphs are encoded to the form of incidence matrix. The size of a hypergraph with  $n$  vertices and  $m$  edges becomes  $O(mn)$  in this coding.

The vector assignment problem of hypergraphs is shown to be in NP by the following reason. A nondeterministic Turing machine (NDTM) first guesses a vector assignment. That is, it guesses whether each element of vector  $v_i$  assigned to vertex  $x_i$  is 0 or 1. This process requires  $O(k)$  time for one vertex, and total  $O(kn)$  time for all vertices. Therefore it takes  $O(n^2)$  time to carry out this process because  $k \leq n$ . The NDTM next checks whether the vector assignment satisfies the condition of linear independence at every edge. In general,  $p(\leq k)$  pieces of  $k$ -dimensional vectors are linearly independent if and only if the rank of the  $k$  by  $p$  matrix obtained by setting those vectors as row vectors is  $p$ . So the NDTM can check this condition in polynomial time of  $k, m$  and  $p$  using Gaussian elimination. As  $p \leq k \leq n$ , the time needed for this check is in polynomial order of  $m$  and  $n$ . Therefore the vector assignment problem of hypergraphs is in NP.

Next it is shown that the coloring problem of graphs is polynomially transformable to the vector assignment problem of hypergraphs. For the problem whether a graph  $G=(X, E)$  is  $k$ -colorable, we construct the following graph  $G'$ , where  $k$  is represented as  $2^{a-1}+b$  ( $0 \leq b \leq 2^{a-1}$ ,  $a$  and  $b$  are integers.)

$$G' = (X \cup X', E \cup E_1 \cup E_2)$$

$$X' = \{x'_1, x'_2, \dots, x'_{2^a-1-b-1}\}$$

$$E_1 = \{\{x_i, x'_j\} \mid x_i \in X, x'_j \in X'\}$$

$$E_2 = \{\{x'_i, x'_j\} \mid x'_i \in X', x'_j \in X', i \neq j\}$$

Namely,  $G'$  is the graph obtained from  $G$  by adding new vertices up to  $2^a-1$  and forming new edges of order two for every pair of old and new vertices and for every pair of new ones.  $G$  is  $k$ -colorable if and only if  $G'$  is  $(2^a-1)$ -colorable because new distinct  $(2^{a-1}-b-1)$  colors are necessary to color the vertices contained in  $X'$ .

$G'$  is  $(2^a-1)$ -colorable if and only if  $G'$  is  $a$ -vector assignable. The reason for this is stated below. If  $G'$  is  $(2^a-1)$ -colorable,  $(2^a-1)$  vectors in  $V^a$  except the zero vector can be assigned to each vertex of  $G'$  so that same vectors are assigned to the vertices to which same colors are assigned. This vector assignment satisfies the condition of linear independence, so  $G'$  is  $a$ -vector assignable. If  $G'$  is  $a$ -vector assignable, colors can be assigned to each vertex of  $G'$  so that same colors are

signed to the vertices to which same vectors are assigned. This coloring satisfies the coloring condition and the number of colors does not exceed  $2^a - 1$ . So  $G'$  is  $(2^a - 1)$ -colorable. Therefore  $G$  is  $k$ -colorable if and only if  $G'$  is  $a$ -vector assignable.

It takes polynomial time of  $k$  and  $n$  to construct  $G'$  from  $G$ . Since  $k \leq n$ , the coloring problem of graphs is proved to be polynomially transformable to the vector assignment problem of hypergraphs. QED

#### 4.2 Vector Assignment of Perfect Hypergraphs

Vector assignment number  $\delta(H_{n,r})$  of the perfect hypergraph of rank  $r$  with  $n$  vertices is regarded as the upper bound of those of arbitrary hypergraphs of rank  $r$  with  $n$  vertices. We have the following theorem which states the relation between the vector assignment numbers of perfect hypergraphs and the construction of linear codes.

**Theorem 3.**  $H_{n,r}$  is  $k$ -vector assignable if and only if a linear code of length  $n$  with  $k$  check digits whose minimum distance is  $d(\geq r+1)$  can be constructed.

**Proof:** If  $H_{n,r}$  is  $k$ -vector assignable, we can construct a  $k$  by  $n$  parity check matrix  $H_{\text{parity}}$  whose column vectors are those assigned to the vertices of  $H_{n,r}$ . Hence arbitrary sums of  $r$  or less pieces of column vectors in  $H_{\text{parity}}$  can not be the zero vector, because arbitrary  $r$  pieces of vectors assigned to the vertices of  $H$  are linearly independent. Therefore the minimum distance of linear code defined by  $H_{\text{parity}}$  is more than or equal to  $r+1$ .

On the contrary, suppose that there exists a linear code of length  $n$  with  $k$  check digits whose minimum distance is  $d(\geq r+1)$ . When all vertices of  $H_{n,r}$  are assigned distinct column vectors of the parity check matrix which defines the linear code, arbitrary sums of  $r$  or less pieces of these vectors can not be the zero vector, because the minimum distance of the linear code is  $d(\geq r+1)$ . Therefore this assignment satisfies the condition of linear independence at every edge and  $H_{n,r}$  is proved to be  $k$ -vector assignable. QED

For locally exhaustive testing for all the subsets of inputs of size  $r$ , a testing scheme using condensed LFSR is proposed in [15]. The condensed LFSR testing is considered to utilize only a sub-class of linear code. Therefore, our testing scheme is more general and can generate more compact test sequences.

In the field of coding theory, the problem whether a linear code can be constructed with a given set of length, number of check digits and minimum distance is solved only in limited cases. Theorem 3 shows that the results offered in coding theory are also true for the vector assignment of perfect hypergraphs. Helgert and Stinaff calculated the upper and lower bounds of binary linear codes for code length less than 128 [14]. By using their results, the upper and lower bounds of vector assignment number  $\delta(H_{127,r})$  are obtained as shown in Fig. 6. For other values of  $n$  less than 128, graphs of the upper and lower bounds of vector assignment number

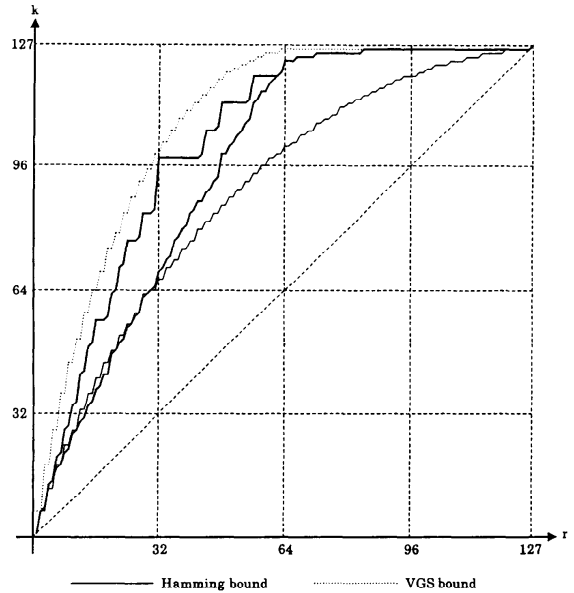


Fig. 6 Upper and lower bound of vector assignment number  $\delta(H_{127,r})$  with Hamming and VGS bound (Data extracted from [14]).

$\delta(H_{n,r})$  have almost the same shape as in the case of  $n=127$ .

Furthermore, we already know the following facts in the coding theory, where  $N(k, r+1)$  denotes the length of the longest linear code under given number of check digits  $k$  and minimum distance  $(r+1)$ .

- 1) When  $r+1=3$ ,  $N(k, 3)=2^k - 1$ .
- 2) When  $r+1=4$ ,  $N(k, 4)=2^{k-1}$ .

By applying the results of Theorem 3, we have the following properties concerning with the vector assignment number of hypergraphs (where  $[x]$  is the smallest integer not smaller than  $x$ ).

Corollary 1.

- 1)  $\delta(H_{n,2}) = [\log_2(n+1)]$ .
- 2)  $\delta(H_{n,3}) = [\log_2 n] + 1$ .

#### 4.3 Graph-Theoretical Properties

Since the vector assignment problem of hypergraphs is proved to be NP-complete, algorithms to solve this problem should try to reduce the size of hypergraphs by some way if possible rather than try to treat large hypergraphs directly. The following graph-theoretical properties can be utilized for this purpose.

**Theorem 4.** If the reduced form of a hypergraph  $H=(X, E)$  is  $H'=(X, E')$ , then  $\delta(H)=\delta(H')$ .

**Proof:** Obvious from the definition of  $\delta(H)$ . QED

**Theorem 5.** If there is a vertex  $x_j$  contained only in one edge  $e_i$  in a hypergraph  $H=(X, E)$ , then  $\delta(H) = \max \{ \delta(H_{X-\{x_j\}}), |e_i| \}$ , where  $H_{X-\{x_j\}}$  is the sub-hypergraph of  $H$  generated by the set  $X - \{x_j\}$ .

**Proof:** Suppose that  $e_i$  contains  $p$  vertices  $x_j, x_{i_1}, x_{i_2}, \dots, x_{i_{p-1}}$  and that  $\delta(H_{X-\{x_j\}})$  is  $k$ . Since  $H_{X-\{x_j\}}$  has an edge



Table 1 Length of the test sequences for the circuit  $C_{n,r}$ .

method	length	
	$C_{n,2}$	$C_{n,3}$
(1a) Constructive method <i>A</i>	$1.893 \cdot \log_2 n + 1^*$	$2 \cdot (\log_2 n)^{1.585} + 2^*$
(1b) Constructive method <i>B</i>	$\log_2 n + 0.5 \cdot \log_2 \log_2 n + O(1)^*$	$0.5 \cdot [\log_2 n] + O(\log_2 n \cdot \log_2 \log_2 n)^*$
(1c) Constant weight vector method	$n + 1$	$2n$
(1d) Method using LFSR	$2^{\lceil \log_2(n+1) \rceil}$	$2 \cdot 2^{\lceil \log_2 n \rceil}$
(2) Verification testing	$n + 1$	$2n$
(3) <i>Proposed method</i>	$2^{\lceil \log_2(n+1) \rceil}$	$2 \cdot 2^{\lceil \log_2 n \rceil}$
(4) Exhaustive testing	$2^n$	$2^n$

\*only for sufficiently large  $n$

$$D = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

(a) Dependence matrix

$$H = (X, E)$$

$$X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$$

$$E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9\}$$

$$e_1 = \{x_1, x_2, x_5, x_6\}$$

$$e_2 = \{x_1, x_3, x_4, x_5\}$$

$$e_3 = \{x_1, x_6, x_7\}$$

$$e_4 = \{x_1, x_2, x_8\}$$

$$e_5 = \{x_2, x_3, x_7\}$$

$$e_6 = \{x_2, x_4, x_6\}$$

$$e_7 = \{x_3, x_4, x_6\}$$

$$e_8 = \{x_3, x_4, x_8\}$$

$$e_9 = \{x_4, x_5, x_7\}$$

$$v_1 = (1000)$$

$$v_2 = (0100)$$

$$v_3 = (1100)$$

$$v_4 = (0011)$$

$$v_5 = (0010)$$

$$v_6 = (0001)$$

$$v_7 = (0110)$$

$$v_8 = (0010)$$

$$\delta(H) = 4$$

(b) An assignment of vectors to the vertices of  $H$ 

Fig. 8 An example of locally exhaustive testing using linear logic circuits. (a) Dependence matrix, (b) An assignment of vectors to the vertices of  $H$ .

sequences. The dependence matrix of the circuit under test is denoted by  $D$  and its corresponding hypergraph is denoted by  $H$ . Here methods of locally exhaustive testing are divided into three categories.

The first method is the test with a locally exhaustive test sequence for all the subsets of inputs of size  $r$ , where  $r$  is the rank of  $H$  [3]–[6]. In these methods, the length of test sequences is determined only by the number of the vertices in  $H$  and the rank of  $H$ . The second method is the verification testing proposed in [8], which utilizes the input-output dependence information of the circuits under test. The last one is the locally exhaustive testing proposed in this paper, in which the vector assignment problem of hypergraphs is solved for the hypergraph  $H$  and the length of the test sequence becomes  $2^{\delta(H)}$ .

The two following examples show that the test se-

Table 2 Length of the test sequences for the circuit  $C$ .

method	length
(1a) Constructive method <i>A</i>	36
(1b) Constructive method <i>B</i>	68
(1c) Constant weight vector method	36
(1d) Method using LFSR	64
(2) Verification testing	28
(3) <i>Proposed method</i>	16
(4) Exhaustive testing	256

quences obtained by our method are shorter in some cases and longer in other cases, compared with those obtained by other methods.

**Example 4.** Table 1 shows the comparison of the length of the test sequences obtained by the following methods for the circuit  $C_{n,r}$  whose dependence matrix is  $H_{n,r}$ .

1) Test with a locally exhaustive test sequence for all the sets of inputs of size  $r$

a) *A* constructive method *A* [6]

b) *A* constructive method *B* [3]

c) The constant weight vector method [5]

d) The method using linear feedback shift registers [4]

2) Verification testing [8]

3) The method using linear logic circuits

4) Exhaustive Testing

Although two constructive methods seem to offer very short test sequences, this is an asymptotic estimation and they work well only for large  $n$ . In this case, test sequences obtained by our method are longer than those by other methods.

**Example 5.** Suppose that locally exhaustive testing is to be performed on the circuit  $C$  whose dependence matrix  $D$  is illustrated in Fig. 8(a). For the hypergraph  $H$  whose incidence matrix is  $D$ , a vector assignment is performed, the result of which is shown in Fig. 8(b). Since  $\delta(H) \geq (\text{rank of } H) = 4$  from Lemma 1, so  $\delta(H) = 4$ . Therefore the length of the test sequence is  $2^4 = 16$  and it realizes the lower bound of the test sequence length for locally exhaustive testing of  $C$ .

Comparison on the length of test sequences for this example is shown in Table 2, where our method gives the shortest test sequence.

Next we give a brief discussion on the cost for generator of test sequences. Among the methods listed

above, test sequences are systematically generated except two constructive methods [3][6]. In [5][8], a constant weight vector generator is necessary. In locally exhaustive testing using linear logic circuits, only an LFSR and an LCC are necessary and test sequence generators are much more compact than for the other methods stated above. From the point of necessary hardware, the method of [4] and exhaustive testing are the best, because only a LFSR is needed.

## 6. Conclusion

In this paper, locally exhaustive testing of combinational circuits using linear logic circuits has been proposed. Locally exhaustive testing is in a sense equivalent to functional testing. It is shown that some class of logic function including important one such as addition with redundant binary number has a property of local computability [13]. Depending on circuit structure, however, the locally exhaustive testing is in some cases insufficient for the test of a logic circuit for some locally computable function. But in most cases such a circuit has the property of local connectivity. Furthermore, many control circuits are apt to be locally connected. In this case, our approach can be directly applied by calculating dependence matrix from circuits structure. The testing scheme proposed in this paper is composed of an LFSR and an LCC. It is designed more compactly than former testing schemes and is suitable for built-in self testing.

The correspondence problem between the inputs of the circuit under test and the outputs of the LCC has been formalized as the vector assignment problem of hypergraphs. This problem resembles the coloring problem of graphs and has been proved to be also NP-complete. Furthermore, it has been shown that the vector assignment problem of perfect hypergraphs is equivalent to the construction problem of linear codes. So generally speaking, the vector assignment problem of hypergraphs is difficult to solve. But using several graph-theoretical properties, the size of hypergraphs can be reduced.

A comparison of the length of test sequences for locally exhaustive testing has been made. Compared with the test sequences offered by other methods, those offered by our method are shorter in some cases and longer in other cases, which depends on the structure of dependence matrix. Our method has a great advantage over other methods when considering the cost for generation of test sequences.

Several problems remain unsolved concerning locally exhaustive testing using linear logic circuits.

1) In order to design test sequence generators compactly, the vector assignment problem of hypergraphs have many variations. For example, the following points should be considered: reduction of the number of modulo 2 adders, use of existent shift registers and so on.

2) If the length of the test sequence for the given cir-

cuit is not acceptable, the circuit should be partitioned into subcircuits. Circuit partitioning techniques should be achieved considering the cost of partitioning.

3) As for the vector assignment problem of hypergraphs, other graph-theoretical properties should be found to reduce the size of hypergraphs. It seems interesting to find the class of hypergraphs whose vector assignment number is equal to or a little larger than their ranks.

## Acknowledgments

The authors are indebted to Professor Yasuo Sugiyama of Setsunan University for providing reference [14]. They also would like to express their sincere appreciation to Dr. H. Yasuura, Mr. N. Takagi, and N. Ishiura of Kyoto University for their invaluable suggestions and discussions. Furthermore, they would like to thank gratefully to all the members of Yajima Laboratory for their discussions.

## References

1. IBARRA, O. H. and SAHNI, S. K. Polynomially Complete Fault Detection Problems, *IEEE Trans. Comput.*, C-24, 3 (March 1975), 242-249.
2. SAEKI, T. and YAJIMA, S. On the Computational Complexity of Test Generation Algorithms for Two-Level Logic Circuits, *Technical Report of IECEJ*, AL81-78 (October 1981).
3. CHANDRA, A. K., KOU, L. T. and MARKOWSKY, G. On Sets of Boolean  $n$ -Vectors with All  $k$ -Projections Surjective, *IBM Res. Report*, RC 8936, (July 1981).
4. TANG, D. T. and CHEN, C. L. Logic Test Pattern Generation Using Linear Codes, *Proc. 13th FTCS*, (June 1983), 222-226.
5. TANG, D. T. and WOO, L. S. Exhaustive Test Pattern Generation with Constant Weight Vectors, *IEEE Trans. Comput.*, C-32, 12, (December 1983), 1145-1150.
6. TANG, D. T., and CHEN, C. L. Iterative Exhaustive Pattern Generation for Logic Testing, *IBM J. Res. & Dev.*, 28, 2 (March 1984), 212-219.
7. LEMPEL, A. and COHN, M. Design of Universal Test Sequences for VLSI, *IEEE Trans. Inform. Theory*, IT-31, 1 (January 1985), 10-17.
8. MCCLUSKEY, E. J. Verification Testing—A Pseudo-exhaustive Test Technique, *IEEE Trans. Comput.*, C-33, 6 (June 1984), 541-546.
9. BERGE, C. Graphs and Hypergraphs, North-Holland (1973), 289-396.
10. PETERSON, W. W. and WELDON, E. J. Error Correcting Codes, 2nd Ed., *MIT Press, Cambridge* (1972).
11. AHO, A. V., HOPCROFT, J. E. and ULLMAN, J. D. The Design and Analysis of Computer Algorithms, *Addison-Wesley, Reading Mass.* (1974), 363-404.
12. CARR, W. N. and MIZE, J. P. MOS/LSI Design and Application, McGraw-Hill (1972).
13. YASUURA, H., TAKAGI, N. and YAJIMA, S. Redundant Coding for Local Computability, in *Discrete Algorithms and Complexity*, A.P., Orlando (1987).
14. HELGERT, H. J. and STINAFF, R. D. Minimum-Distance Bounds for Binary Linear Codes, *IEEE Trans. Inform. Theory*, IT-19, 3 (May 1973), 344-356.
15. WANG, L. T. and MCCLUSKEY, E. J. Condensed Linear Feedback Shift Register (LFSR) Testing—A Pseudoexhaustive Test Technique, *IEEE Trans. Comput.*, C-35, 4 (April 1986), 367-370.
16. KAWAHARA, K. Locally Exhaustive Testing of Combinational Circuits Using Linear Logic Circuits, Master thesis of Department of Information Science, Kyoto University (February 1985).
17. KAWAHARA, K., HIRAISHI, H. and YAJIMA, S. Locally Exhaustive Testing of Combinational Circuits Using Linear Feedback Shift Registers, *Technical Report of IECEJ*, AL84-61 (January, 1985).

(Received June 26, 1987)