D. T. Tang
C. L. Chen

Iterative Exhaustive Pattern Generation for Logic Testing

Exhaustive pattern logic testing schemes provide all possible input patterns with respect to an output in the set of test patterns. This paper is concerned with the problem that arises when this is to be done simultaneously with respect to a number of outputs, using a single test set. More specifically, in this paper we describe an iterative procedure for generating a test set consisting of n-dimensional vectors which exhaustively covers all k-subspaces simultaneously, i.e., the projections of n-dimensional vectors in the test set onto any input subset of a specified size k contain all possible patterns of k-tuples. For any given k, we first find an appropriate N ($N > k$) *and generate an efficient N-dimensional test set for exhaustive coverage of all k-subspaces. We next develop a constructive procedure to expand the corresponding test matrix (formed by taking test vectors as its rows) such that a test set of* N^2*-dimensional vectors exhaustively covering the same k-subspaces is obtained. This procedure may be repeated to cover arbitrarily large n* ($n = N^{2^i}$ *after i iterations), while keeping the same k. It is shown that the size of the test set obtained this way grows in size which becomes proportional to (log n) raised to the power of* $\lceil \log (q + 1) \rceil$*, where q is a function of k only, and is approximated (bounded closely below) by* $k^2/4$ *in binary cases. This approach applies to nonbinary cases as well except that the value of q in an r-ary case is approximated by a number lying between* $k^2/4$ *and* $k^2/2$*.*

Introduction

In the conventional approach to logic circuit testing, a set of test vectors to be applied at the circuit inputs is derived from an analysis made on the circuit under test while considering a predetermined set of faults to be detected, typically the set of single stuck-at-0 or stuck-at-1 faults at the gate level [1]. Such a test-generation procedure requires a substantial amount of computer time due to the necessary analysis and simulation to be carried out.

As the number of circuits packed onto a VLSI chip grows larger, the tasks involved in the conventional approaches of logic test generation and fault simulation become increasingly difficult. The standard assumption of the single stuck-at-fault model also becomes more inadequate [2]. A partial solution to this problem is to exercise exhaustive pattern testing, with respect to each output, while relying on certain partitioning techniques to limit the size of each input subset associated with an output [3].

With the exhaustive pattern testing approach, the set of all inputs feeding an output is generally provided with all possible input patterns from the test set. Therefore, any single hard fault or combination of hard faults which results in a permanent alteration of the truth table associated with the output function is tested.

Exhaustive pattern testing of logic circuits has several attractive features. In addition to the fact that test patterns can be generated quite easily, the process and its fault coverage are no longer dependent directly on the fault model assumed or on the specific circuit under test. An immediate problem, however, is how to provide exhaustive input patterns simultaneously with respect to many outputs associated with the same circuit. There is a need to develop a basic theoretical understanding with regard to how a single test set can be generated for this purpose and how efficient such test sets can be.

The problem of generating a single set of test vectors to provide simultaneously all possible input patterns to each of a collection of input subsets has been investigated in the past [4, 5]. Recently, two general methods based, respectively, on constant weight and on linear codes have been developed for this problem [6–8]. While these methods give test sets which, due to the simplicity of the mathematical structure used, can be implemented quite easily, the efficiency of such test sets is not always high. Thus, in an attempt to apply these results in practice, one may well find that these exhaustive test-pattern-generation techniques lead to test sets of excessive size even for moderately sized circuits. This seems to imply that exhaustive pattern testing would be impractical unless sufficient logic circuit partitioning is exercised in the design stage. On the other hand, one may consider using the exhaustive pattern testing approach together with other testing methods such as pseudo-random pattern testing. In any case, new techniques for generating more efficient test sets for exhaustive pattern testing would be very desirable.

The objective of this paper is to develop a constructive procedure for building more efficient test sets of higher dimensions from those of lower dimensions. It is shown that test sets generated this way are generally much more efficient (smaller) for large n (dimension of test vectors) compared to test sets generated by previously known methods for the same exhaustive coverage. The asymptotic growth behavior of such test sets thus provides upper bounds to the efficiency of optimal test sets for large n in such an exhaustive pattern testing approach.

Basic concepts

We now briefly summarize some basic concepts and results from our earlier paper [6].

Consider, in general, an r-valued logic circuit ($r \geq 2$) with n inputs forming an r-ary n-space. An output of the logic circuit may be dependent on a subset of k inputs forming a k-subspace (k-dimensional projection) of the n-space. Note that an input may actually represent a cluster of disjoint inputs not connected to the same output [9].

Definition
A *test matrix* is a matrix in which the rows are test vectors. The column dimension is the dimension of the test vectors and the row dimension is the size of the test set.

Definition
A set $\mathbf{T}$ of n-vectors in an r-ary n-space, represented by row vectors of a test matrix $\mathbf{T}$, exhaustively covers a k-subspace if and only if the projection of $\mathbf{T}$ onto this k-subspace contains all r^k distinct r-ary patterns.

Definition
The weight w of an r-ary n-vector $\mathbf{v} = (v_1, v_2, \cdots, v_n)$ is the regular sum of all n components of $\mathbf{v}$. That is, $w = v_1 + v_2 + \cdots + v_n$.

- *Theorem 1*

Given $n > k$ and $r \geq 2$, then $\mathbf{T}$, a set of vectors in the r-ary n-space, exhaustively covers all k-subspaces if it contains all r-ary n-vectors of weight(s) w such that $w = c \bmod s$, where $s = (n-k)(r-1)+1$, and $0 \leq c \leq s-1$.

- *Theorem 2*

Let $\mathbf{T}$ be any set of n-vectors in an r-ary space, which exhaustively covers all k-subspaces; then its minimum size $|\mathbf{T}|_{\min}$ is bounded as follows:

$$r^k \leq |\mathbf{T}|_{\min} \leq \frac{r^n}{(n-k)(r-1)+1}. \tag{1}$$

Theorems 1 and 2 are best illustrated by some examples.

Example 1: n = 20, k = 3, r = 2
From Theorem 1, $s = (n-k)(r-1)+1 = 18$. If we let $w = 1 \bmod 18$, then $w = 1, 19$. Thus $\mathbf{T}_1$ consists of all weight-1 and weight-19 vectors:

$$\mathbf{T}_1 = \left[\begin{array}{ccccc} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & & & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & & & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ \hline 0 & 1 & \cdots & 1 & 1 \\ 1 & 0 & & & 1 \\ \vdots & & \ddots & & \vdots \\ 1 & & & 0 & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{array}\right]$$

Here $|\mathbf{T}_1| = 20 + 20 = 40$.

Seventeen alternative solutions may also be obtained by setting w to values other than 1 modulo 18. The solution $\mathbf{T}_1$ is the one with the smallest size among 18 solutions obtained via Theorem 1.

Example 2: n = 4, k = 2, r = 3
From Theorem 1, $s = (n-k)(r-1)+1 = 5$. If we let $w = 1 \bmod 5$, then $w = 1,6$. Thus, $\mathbf{T}_1$ consists of all weight-1 and weight-6 vectors:

$$\mathbf{T}_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 2 & 2 & 2 \\ 1 & 1 & 2 & 2 \\ 1 & 2 & 1 & 2 \\ 1 & 2 & 2 & 1 \\ 2 & 0 & 2 & 2 \\ 2 & 1 & 1 & 2 \\ 2 & 1 & 2 & 1 \\ 2 & 2 & 0 & 2 \\ 2 & 2 & 1 & 1 \\ 2 & 2 & 2 & 0 \end{bmatrix}$$

Here $|\mathbf{T}_1| = 4 + 10 = 14$.

Four alternative solutions may also be obtained by setting w to values other than 1 modulo 5.

Example 3: $n - k = 1,\ r \geq 2$
Here $s = (n - k)(r - 1) + 1 = r$. There are r disjoint test sets, each of which corresponds to a single parity r-ary code with $|\mathbf{T}| = r^n/r = r^k\ |\mathbf{T}|_{\min}$. For $n = 4$, $k = 3$, $r = s = 2$, $w = 1 \bmod 2$,

$$\mathbf{T}_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}.$$

For $n = 3$, $k = 2$, $r = s = 4$, $w = 0 \bmod 4$,

$$\mathbf{T}_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 3 \\ 0 & 2 & 2 \\ 0 & 3 & 1 \\ 1 & 0 & 3 \\ 1 & 1 & 2 \\ 1 & 2 & 1 \\ 1 & 3 & 0 \\ 2 & 0 & 2 \\ 2 & 1 & 1 \\ 2 & 2 & 0 \\ 2 & 3 & 3 \\ 3 & 0 & 1 \\ 3 & 1 & 0 \\ 3 & 2 & 3 \\ 3 & 3 & 2 \end{bmatrix}.$$

It should be clear from the preceding theorems and examples that a test set generated via Theorem 1 for the exhaustive coverage of all k-subsets is optimal when $k = n - 1$. This can be seen from Theorem 2: With $n - k = 1$, Eq. (1) becomes

$$r^k \leq |\mathbf{T}|_{\min} \leq \frac{r^n}{r} = r^k,$$

and the size reaches the lower bound r^k. When k is close to n, the solution remains near-optimal.

For a fixed k and large n, the size of the test sets obtained from Theorem 1 is minimized when c is chosen to be as close to $k/2$ as possible [6]. Therefore, the test set size becomes proportional to

$$\binom{n}{k/2} \cong n^{k/2}.$$

The test sets obtained from linear codes [7] are generally not any more efficient in terms of their size. The iterative procedure we describe next results in much more efficient test sets whose sizes grow asymptotically within a constant multiplication factor of $(\log n)$ raised to the power of $[\log (q + 1)]$, where q is approximated by a number lying between $k^2/4$ and $k^2/2$.

An iterative procedure

Let $\mathbf{P}(i)$ denote 1-by-N cyclic permutation vectors as follows:

$$\mathbf{P}(1) = (1, 2, 3, \cdots, N - 1, 0),$$

$$\mathbf{P}(2) = (2, 3, \cdots, N - 1, 0, 1),$$

$$\cdots$$

$$\mathbf{P}(N - 1) = (N - 1, 0, 1, \cdots, N - 2).$$

We also denote a constant N-vector by $\mathbf{C}(i)$, $\mathbf{C}(i) = (i, i, \cdots, i)$.

We next consider "composite permutation matrices" which are constructed from the preceding permutation vectors and constant N-vectors. Such a composite permutation matrix can be partitioned into regular blocks of size 1-by-N, each of which is a permutation vector or a constant N-vector.

- *Theorem 3*

Let N be a prime power, i.e., $N = m^i$, where m = prime and $i \geq 1$. Then the following $(m + 1)$ by N^2 composite matrix of the form

$$\mathbf{M} = \begin{bmatrix} \mathbf{C}(0), & \mathbf{C}(1), & \mathbf{C}(2), & \cdots, & \mathbf{C}(N-1) \\ \mathbf{P}(0), & \mathbf{P}(0), & \mathbf{P}(0), & \cdots, & \mathbf{P}(0) \\ \mathbf{P}(0), & \mathbf{P}(1), & \mathbf{P}(2), & \cdots, & \mathbf{P}(N-1) \\ \mathbf{P}(0), & \mathbf{P}(2), & \mathbf{P}(4), & \cdots, & \mathbf{P}[2(N-1)] \\ \cdots & & & & \\ \cdots & & & & \\ \mathbf{P}(0), & \mathbf{P}(m-1), & \mathbf{P}[2(m-1)], & \cdots, & \mathbf{P}[(N-1)(m-1)] \end{bmatrix} \bmod N \tag{2}$$

does not have any 2-by-2 submatrix with identical columns.

Proof

In the preceding matrix $\mathbf{M}$, N^2 columns are naturally partitioned into N groups of N columns each. The theorem is clearly true for any 2-by-2 submatrix taken out of any single group, since at least one of its two rows must come from some

P(i) which has distinct elements (integer components) and thus two columns cannot be the same. Now consider any two 2-by-2 submatrix taken out of rows s and t ($t > s$), and column groups j and k ($k > j$). If s is not the top row, the submatrix must be of the form

$$\begin{bmatrix} js + b & ks + c \\ jt + b & ki + c \end{bmatrix} \bmod N.$$

Assume that the submatrix has two identical columns. Then $(k - j)(t - s) \bmod N$ must be equal to zero. To see that this cannot be true, we note that $(k - j) \le N$ and $(t - s) \le m$ (the latter being true since $t,s > 1$). Thus, $(k - j)(t - s)$ cannot contain a factor of N, $N = m^i$. This means that the elements in any 2-by-2 submatrix cannot be identical (mod N) in both rows, and hence the theorem. Q.E.D.

We now consider a test matrix $\mathbf{T}(N^2)$ which is constructed according to the composite permutation matrix **M** of Theorem 3. This new test matrix is "grown" from the base matrix $\mathbf{T}(N)$ which exhaustively covers all k-subspaces. If we denote the size of the test set corresponding to the base matrix by B, $B = |\mathbf{T}(N)|$, then this new text matrix results in a test set of size $(m + 1)B$, consisting of N^2-dimensional vectors.

Example 4
Consider a basic matrix $\mathbf{T}(N)$ as follows:

$$\mathbf{T}(3) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}. \tag{3}$$

Since this matrix is generated according to Example 3, it exhaustively covers all 2-subspaces. The composite permutation matrix $\mathbf{M}(N^2)$ as described in Theorem 3 is

$$\mathbf{M}(9) = \begin{bmatrix} (0\ 0\ 0) & (1\ 1\ 1) & (2\ 2\ 2) \\ (0\ 1\ 2) & (0\ 1\ 2) & (0\ 1\ 2) \\ (0\ 1\ 2) & (1\ 2\ 0) & (2\ 0\ 1) \\ (0\ 1\ 2) & (2\ 0\ 1) & (1\ 2\ 0) \end{bmatrix}. \tag{4}$$

It is not difficult to check and see that there is no 2-by-2 submatrix in **M**(9) with identical columns. To "grow" the new test matrix **T**(9) from **T**(3), we merely replace each entry in **M**(9), considered as a column index, with the corresponding column in the base matrix **T**(3):

$$\mathbf{T}(9) = \left[\begin{array}{ccc} 0\,0\,0 & 0\,0\,0 & 0\,0\,0 \\ 0\,0\,0 & 1\,1\,1 & 1\,1\,1 \\ 1\,1\,1 & 0\,0\,0 & 1\,1\,1 \\ 1\,1\,1 & 1\,1\,1 & 0\,0\,0 \\ \hline 0\,0\,0 & 0\,0\,0 & 0\,0\,0 \\ 0\,1\,1 & 0\,1\,1 & 0\,1\,1 \\ 1\,0\,1 & 1\,0\,1 & 1\,0\,1 \\ 1\,1\,0 & 1\,1\,0 & 1\,1\,0 \\ \hline 0\,0\,0 & 0\,0\,0 & 0\,0\,0 \\ 0\,1\,1 & 1\,1\,0 & 1\,0\,1 \\ 1\,0\,1 & 0\,1\,1 & 1\,1\,0 \\ 1\,1\,0 & 1\,0\,1 & 0\,1\,1 \\ \hline 0\,0\,0 & 0\,0\,0 & 0\,0\,0 \\ 0\,1\,1 & 1\,0\,1 & 1\,1\,0 \\ 1\,0\,1 & 1\,1\,0 & 0\,1\,1 \\ 1\,1\,0 & 0\,1\,1 & 1\,0\,1 \end{array}\right]. \tag{5}$$

Let us denote by $\mathbf{T}_1(9)$ the row subspace consisting of the first four rows of **T**(9) corresponding to the first row of **M**(9). Consider the 2-subspace **S**(2) in $\mathbf{T}_1(9)$ consisting of the first and the last columns. Since the corresponding column indices in **M**(9) are 0 and 2, **S**(2) is the same as the 2-subspace in **T**(3) consisting of the first and the last columns there. But the base matrix **T**(3) exhaustively covers all of its 2-subspaces; thus, no binary 2-tuple could be missing from **S**(2). On the other hand, consider the 2-subspace $\mathbf{S}'(2)$ in $\mathbf{T}_1(9)$ consisting of the first two columns. Since the corresponding column indices in **M**(9) are both 0, $\mathbf{S}'(2)$ is the same as the first column in **T**(3) duplicated. Clearly, any binary pattern specifying different values in these two columns, namely (0 1) or (1 0), cannot appear in $\mathbf{S}'(2)$.

In general, the new test matrix $\mathbf{T}(N^2)$ being grown from the base matrix $\mathbf{T}(N)$ according to the composite matrix **M** is characterized in the following theorem.

- *Theorem 4*

Consider an arbitrary k-subspace, $\mathbf{S}(k)$, in $\mathbf{T}(N^2)$. An r-ary k-tuple, $t(k)$, is missing in $\mathbf{S}(k)$ if and only if the following is true in each row of **M**.

Condition A Out of k columns specified by $\mathbf{S}(k)$, there exist two at which the row in **M** assumes identical values but $t(k)$ assumes distinct values.

To prove the "if" part, we observe that identical elements in **M** lead to identical elements in $\mathbf{T}(N^2)$ within the row subspace corresponding to the row in **M** under consideration. Clearly, if Condition A holds, $t(k)$ must differ at least one position from each k-tuple in this part of $\mathbf{T}(N^2)$. Since the same situation holds for all rows of **M**, $t(k)$ must be missing from $\mathbf{S}(k)$ in $\mathbf{T}(N^2)$.

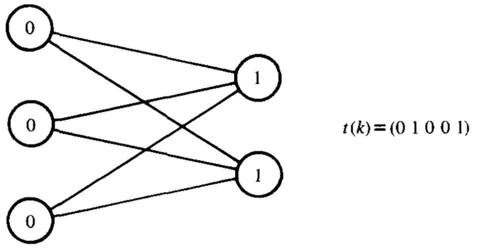

Figure 1 The complete bipartite graph corresponding to $t(k)$.

To prove the "only if" part, let us assume that Condition A does not hold for some row in **M**. Corresponding to this row of **M**, consider a k'-subspace $\mathbf{S}'(k')$ in this part of $\mathbf{T}(N^2)$, obtained from $\mathbf{S}(k)$ by keeping one column position from each group with identical elements. Let $t'(k')$ be the projection of $t(k)$ onto $\mathbf{S}'(k')$; then $t'(k')$ cannot be missing from $\mathbf{S}'(k')$ since $k' \leq k$, and the corresponding subspace in $\mathbf{T}(k)$ is exhaustively covered. Clearly, if Condition A does not hold, by duplicating columns corresponding to identical indices, $t(k)$ can be reconstructed in $\mathbf{S}(k)$ from $\mathbf{S}'(k')$. It follows that $t(k)$ cannot be missing from $\mathbf{T}(N^2)$. Q.E.D.

We next prove our main theorem for the binary case.

- *Theorem 5*

In the binary case, let k $(k > 1)$ be given. If a base matrix $\mathbf{T}(N)$ exhaustively covering all of its k-subspaces is found with $N = m^i$ such that $m =$ prime, $i \geq 1$, $N \geq k$, and $m \geq \lfloor k^2/4 \rfloor$ ($\lfloor x \rfloor$ denotes the largest integer not greater than x), then the new composite test matrix $\mathbf{T}(N^2)$ constructed according to any combinations of $\lfloor k^2/4 \rfloor + 1$ rows of **M** exhaustively covers all of its k-subspaces.

Proof

Suppose that a binary k-tuple, $t(k)$, is missing from a certain k-subspace, $\mathbf{S}(k)$, in $\mathbf{T}(N^2)$. The columns in $\mathbf{S}(k)$ are thus partitioned into two groups according to the values they intercept in $t(k)$. We may construct a complete bipartite graph [10] of k vertices, where these k vertices correspond to the columns in $\mathbf{S}(k)$ and are colored 0 or 1 according to their corresponding values in $t(k)$. This is shown in **Figure 1.** Now if $t(k)$ is missing from $\mathbf{S}(k)$, then from Theorem 4, each row of **M** contains some 0-1 column pair with identical elements. Each such pair, represented by an edge in this graph, is contained in at most one column group of identical projected values from a certain row in **M**, because otherwise this would result in a 2-by-2 submatrix of **M** with identical columns, violating Theorem 3. This means that the number of rows in **M** must be no more than the number of edges in the complete bipartite graph. But the maximum number of edges in any bipartite graph of k vertices can be shown to be $\lfloor k^2/4 \rfloor$ and yet **M** has at least $\lfloor k^2/4 \rfloor + 1$ rows from our construction conditions of **M**, which is a contradiction. It follows that no k-tuple can be missing from any k-subspace in $\mathbf{T}(N^2)$. Q.E.D.

It can be seen from the preceding theorem that, as long as the base matrix $\mathbf{T}(N)$ satisfies the conditions required, it is immaterial how $\mathbf{T}(N)$ is actually obtained. Furthermore, if the size of the base matrix is B, i.e., $B = |\mathbf{T}(N)|$, then the size of the new test matrix is $|\mathbf{T}(N^2)| = B(q + 1)$, where $q = \lfloor k^2/4 \rfloor$ in the binary case. If we were to iterate this procedure j times, we would obtain a test matrix $\mathbf{T}(n)$ such that

$$n = (N^{2^j}) \tag{6}$$

$$\text{and } |\mathbf{T}(n)| = B(q + 1)^j. \tag{7}$$

Observing that $(2^j) = (\log n)/(\log N)$ where $\log x$ is of base 2, we have

$$\begin{aligned} |\mathbf{T}(n)| &= B(q + 1)^j \\ &= B[2^{\log(q+1)}]^j \\ &= B(2^j)^{\log(q+1)} \\ &= B\left(\frac{\log n}{\log N}\right)^{\log(q+1)} \end{aligned} \tag{8}$$

Therefore, the size of the test set obtained by applying Theorem 5 iteratively is proportional to $\log n$ raised to the power of $\log(q + 1)$, where q is a function of k only [roughly $2(\log k)$] and is independent of the base test matrix $\mathbf{T}(N)$. The coefficient, $B/\log N^{\log(q+1)}$ may be minimized by the selection of $\mathbf{T}(N)$.

A straightforward way to select a reasonable set of matrix parameters is as follows: For any given coverage range k, let $N = m \geq k$, and $m =$ the smallest prime $\geq q$. In general, N and m may be selected in various ways as long as the conditions stated in Theorem 5 are satisfied. This is shown in the following examples.

Example 5

Consider $k = 2$ for the binary case, for which $q = 1$. We may select $N = m = 3$ and generate the base matrix $\mathbf{T}(N)$ as in Eq. (3) of Example 4:

$$\mathbf{T}(3) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

We may take the first two rows of $\mathbf{M}(N^2)$, shown in Eq. (4), according to Theorem 3:

$$\mathbf{M}(9) = \begin{bmatrix} (0\ 0\ 0) & (1\ 1\ 1) & (2\ 2\ 2) \\ (0\ 1\ 2) & (0\ 1\ 2) & (0\ 1\ 2) \end{bmatrix}.$$

To obtain the new test matrix T(9), we merely replace each entry in M(9), which is a column index, with the corresponding column in T(3):

$$\mathbf{T}(9) = \left[\begin{array}{ccc} 0\,0\,0 & 0\,0\,0 & 0\,0\,0 \\ 0\,0\,0 & 1\,1\,1 & 1\,1\,1 \\ 1\,1\,1 & 0\,0\,0 & 1\,1\,1 \\ 1\,1\,1 & 1\,1\,1 & 0\,0\,0 \\ \hline 0\,0\,0 & 0\,0\,0 & 0\,0\,0 \\ 0\,1\,1 & 0\,1\,1 & 0\,1\,1 \\ 1\,0\,1 & 1\,0\,1 & 1\,0\,1 \\ 1\,1\,0 & 1\,1\,0 & 1\,1\,0 \end{array}\right].$$

This completes the first iteration. For the second iteration, we would take T(9) as our base matrix and construct M(81) and T(81) in the same fashion as before. When this procedure is repeated j times, we have

$$n = 3^{2^j}$$

and

$$|\mathbf{T}(n)| = 4^{2^j},$$

or

$$|\mathbf{T}(n)| = \frac{4 \log n}{\log 3}.$$

Note that the all-zeros vector in T(9) is repeated. When such repeated all-zeros vectors are removed from T(9) as well as from test matrices T(n) of succeeding iterations, we have

$$|\mathbf{T}(n)| = 1 + 3(2^j),$$

or

$$|\mathbf{T}(n)| = 1 + 3(\log n)/(\log 3) = 1 + 1.893(\log n).$$

Example 6

Consider $k = 3$ for the binary case, for which $q = 2$. We may select $N = 2^2 = 4$ with $m = 2$, and generate the base matrix T(N) according to Example 3:

$$\mathbf{T}(4) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

The composite permutation matrix M(16) is as follows:

$$\mathbf{M}(16) = \begin{bmatrix} (0\;0\;0\;0) & (1\;1\;1\;1) & (2\;2\;2\;2) & (3\;3\;3\;3) \\ (0\;1\;2\;3) & (0\;1\;2\;3) & (0\;1\;2\;3) & (0\;1\;2\;3) \\ (0\;1\;2\;3) & (1\;2\;3\;0) & (2\;3\;0\;1) & (3\;0\;1\;2) \end{bmatrix}.$$

To obtain T(16), we replace each entry in M(16) with the corresponding column it represents in T(4). T(16) is 24 by 16. For the second iteration, we treat T(16) as the base matrix and construct M(256) and subsequently T(256) in a similar fashion. After j iterations, we have

$$n = 4^{2^j}$$

and

$$|\mathbf{T}(n)| = 8(3^j),$$

or

$$|\mathbf{T}(n)| = (8/3)(\log n)^{\log 3}.$$

Here the all-zeros and all-ones vectors are repeated in the test matrices T(n). When such repeated vectors are removed, we have

$$|\mathbf{T}(n)| = 2 + 6(3^j),$$

or

$$|\mathbf{T}(n)| = 2 + 2(\log n)^{\log 3} = 2 + 2(\log n)^{1.585}.$$

This result is better than that obtained in Ref. [4], in which the test set size for $k = 3$ grows asymptotically proportional to $(\log n)^2$.

Example 7

Consider $k = 4$ for the binary case with $q = 4$. We may select $N = m = 5$ and generate the base matrix T(5) according to Theorem 1: For $s = N - k + 1 = 2$, let $w = 0 \bmod 2$. This gives a T(5) which is 16 by 5, consisting of even-weight 5-tuples:

$$\mathbf{T}(5) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}.$$

The required composite permutation matrix M(25) may be obtained by taking any $q + 1 = 5$ rows from Eq. (2):

$$\mathbf{M}(25) = \begin{bmatrix} (01234) & (01234) & (01234) & (01234) & (01234) \\ (01234) & (12340) & (23401) & (34012) & (40123) \\ (01234) & (23401) & (40123) & (12340) & (34012) \\ (01234) & (34012) & (12340) & (40123) & (23401) \\ (01234) & (40123) & (34012) & (23401) & (12340) \end{bmatrix}.$$

Substituting for entries of $\mathbf{M}(25)$ the corresponding columns in $\mathbf{T}(5)$, we obtain $\mathbf{T}(25)$, which is 80 by 25. Again, the repeated all-zeros vectors can be eliminated to reduce $\mathbf{T}(25)$ to 76 by 25. After j iterations, we have

$$n = 5^{2^j}$$

and

$$|\mathbf{T}(n)| = 1 + 15(5^j),$$

or

$$|\mathbf{T}(n)| = 1 + \frac{15}{(\log 5)^{\log 5}} (\log n)^{\log 5}$$

$$= 1 + 2.12 (\log n)^{2.32}.$$

This is better than the result in Ref. [4], in which the test set size for $k = 4$ grows asymptotically proportional to $(\log n)^3$.

Example 8

Consider $k = 5$ for the binary case, for which $q = 6$. We may select $N = m = 7$ and generate the base matrix $\mathbf{T}(7)$ according to Theorem 1: For $s = N - k + 1 = 3$, let $w = 1 \bmod 3$, or $w = 1,4$. $\mathbf{T}(7)$ thus generated contains 7 vectors of weight one and 35 vectors of weight four, giving $B = 42$. The $q + 1 = 7$ rows of the composite permutation matrix $\mathbf{M}(49)$ may be obtained from Eq. (2). Substituting for entries of $\mathbf{M}(49)$ the corresponding columns in $\mathbf{T}(7)$, we obtain $\mathbf{T}(49)$, which is 294 by 49. After this procedure is iterated j times, we have

$$n = 7^{2^j}$$

and

$$|\mathbf{T}(n)| = 42(7^j),$$

or

$$|\mathbf{T}(n)| = \frac{42(\log n)^{\log 7}}{(\log 7)^{\log 7}} = 2.31 (\log n)^{2.8}.$$

Theorem 5 can be generalized to cover the nonbinary cases, as shown in the following theorem.

• *Theorem 6*

In a general r-ary case, let k $(k > 1)$ be given. If a base matrix $\mathbf{T}(N)$ exhaustively covering all of its k-subspaces is found with $N = m^i$ such that m = prime, $i \geq 1$, $N \geq k$, and $m \geq q(r,k)$, where $q(r,k)$ is the maximum number of edges possible in a graph of k vertices and chromatic index r [10], then the new composite test matrix $\mathbf{T}(N^2)$ constructed according to $q(r,k) + 1$ rows of $\mathbf{M}$ exhaustively covers all of its k-subspaces.

Proof

The proof of this theorem is similar to that of Theorem 5. Suppose that an r-ary k-tuple, $t(k)$, is missing from a certain k-subspace, $\mathbf{S}(k)$, in $\mathbf{T}(N^2)$. We may construct a complete r-chromatic graph of k vertices, where these k vertices correspond to the columns in $\mathbf{S}(k)$ and are colored $0, 1, 2, \cdots, (r - 1)$ according to their corresponding values in $t(k)$. From Theorem 4, each row in $\mathbf{M}$ contains some pair of vertices with distinct colors but identical elements. From Theorem 3, each pair is contained in at most one row of $\mathbf{M}$. This means that the number of rows in $\mathbf{M}$ must be no more than $q(r,k)$, the maximum number of edges possible in a graph of k vertices and chromatic index r, which violates our construction condition of $\mathbf{M}$, namely $m \geq q(r,k)$. It follows that no k-tuple can be missing from any k-subspace in $\mathbf{T}(N^2)$. Q.E.D.

The value of $q(r,k)$ for $k \geq 2$ is clearly bounded as follows:

$$\lfloor k^2/4 \rfloor \leq q(r,k) \leq k(k-1)/2 \leq \lfloor k^2/2 \rfloor - 1. \qquad (9)$$

The exact value of $q(r,k)$ is the number of edges in a complete r-chromatic graph of k vertices with as nearly equal numbers of vertices in r color classes as possible. It can be shown [10] that

$$q(r,k) = [k(k-1)/2] - (\lfloor k/r \rfloor - 1)[k - (r/2)\lfloor k/r \rfloor]. \qquad (10)$$

For $r < 8$, a simpler form can be shown to hold:

$$q(r,k) = \lfloor (k^2)(r-1)/2r \rfloor \quad \text{for } 1 < r < 8. \qquad (11)$$

Values of $q(r,k)$ for $r < 8$ and $k < 14$ are listed in **Table 1.**

It is interesting to note that when the procedure described in Theorem 6 is iterated j times in an r-ary case, the dimension n and the test set size $|\mathbf{T}(n)|$ change exactly the same way as shown in Eqs. (6) and (8) in the binary case. That is, asymptotically for large n, $|\mathbf{T}(n)|$ is still proportional to $(\log n)$ raised to the power of $\log(q + 1)$, as shown in Eq. (8), except that q should be interpreted as $q(r,k)$. The selection of matrix parameters in r-ary cases is essentially the same as that in the binary case.

Conclusion

We have described in this paper a constructive procedure for generating iteratively test sets of large dimension n which exhaustively cover all k-subspaces simultaneously. The size of test sets obtained by applying such a constructive procedure iteratively becomes asymptotically proportional to $\log n$ raised to the power of $\log(q + 1)$, where q is a function of k only, bounded closely below by $k^2/4$. The same approach has been shown to be also applicable to nonbinary cases (where base $r > 2$) with similar results except that q is a function of k and the base r. We have shown that $q(r,k)$ is the maximum number of edges possible in a graph of k vertices and chromatic index r, and it is bounded below by a number lying between $k^2/4$ and $k^2/2$.

The results presented in this paper on the asymptotic behavior of the size of test sets generated for exhaustive coverage show a significant improvement over previous methods [4, 6–8]. The iterative procedure also enables one to build test sets of higher dimensions from those of lower ones which may be generated by other means and found to be optimal or near-optimal.

Unless a sufficient degree of partitioning of logic circuits is built into the design procedure, the best chance for the exhaustive testing technique as described in this paper may be its use together with other testing approaches, such as the pseudo-random pattern testing approach, in which case the degree of exhaustive coverage k may be limited to a practical value.

References

1. E. E. Muehldorf and A. D. Savkar, "LSI Logic Testing—An Overview," *IEEE Trans. Computers* **C-30,** 1–17 (January 1981).
2. V. K. Agarwal and A. S. F. Fung, "Multiple Fault Testing of Large Circuits by Single Fault Test Sets," *IEEE Trans. Computers* **C-30,** 855–865 (November 1981).
3. E. J. McCluskey and S. Bozorgui-Nesbat, "Design for Autonomous Test," *IEEE Trans. Computers* **C-30,** 866–875 (November 1981).
4. A. K. Chandra, L. T. Kou, and G. Markowsky, "On Sets of Boolean *n*-Vectors with all *k*-Projections Surjective," *Research Report RC-8936,* IBM Thomas J. Watson Research Center, Yorktown Heights, NY, July 1981.
5. Z. Barzilai, D. Coppersmith, and A. Rosenberg, "Exhaustive Generation of Bit Patterns with Applications to VLSI Testing," *IEEE Trans. Computers* **C-32,** 190–194 (February 1983).
6. D. T. Tang and L. S. Woo, "Exhaustive Test Pattern Generation with Constant Weight Vectors," *Research Report RC-9442* IBM Thomas J. Watson Research Center, Yorktown Heights, NY, June 1982. Also published in *IEEE Trans. Computers* **C-32,** 1145–1150 (December 1983).
7. D. T. Tang and C. L. Chen, "Logic Test Pattern Generation Using Linear Codes," *Proceedings of the 13th International Conference on Fault-Tolerant Computing,* Milan, Italy, June 1983, pp. 222–226.
8. E. J. McCluskey, "Built-In Verification Test," *Digest of Papers,* 1982 International Test Conference, Cherry Hill, PA, November 16–18, 1982, pp. 183–190.
9. Z. Barzilai, J. Savir, G. Markowsky, and M. G. Smith, "VLSI Self-Testing Based on Syndrome Techniques," *Digest of Papers, 1981 International Test Conference,* Cherry Hill, PA, October 1981, pp. 102–109.
10. C. Berge, *Graphs and Hypergraphs,* North-Holland Publishing Company, Amsterdam-London, 1973.

Table 1 Values of $q(r,k)$ for $1 < r < 8$ and $k < 14$.

k	$q(2,k)$	$q(3,k)$	$q(4,k)$	$q(5,k)$	$q(6,k)$	$q(7,k)$
1	0	0	0	0	0	0
2	1	1	1	1	1	1
3	2	3	3	3	3	3
4	4	5	6	6	6	6
5	6	8	9	10	10	10
6	9	12	13	14	15	15
7	12	16	18	19	20	21
8	16	21	24	25	26	27
9	20	27	30	32	33	34
10	25	33	37	40	41	42
11	30	40	45	48	50	51
12	36	48	54	57	60	61
13	42	56	63	67	70	72

Received June 29, 1983; revised October 19, 1983

C. L. (Jim) Chen *IBM Data Systems Division, P.O. Box 390, Poughkeepsie, New York 12602.* Dr. Chen is a senior engineer working on error-correcting codes and fault-tolerant memory systems. Before joining IBM in 1974, he held a postdoctoral position at the University of Hawaii and was a faculty member of the University of Illinois. He received his Ph.D. degree in electrical engineering from the University of Hawaii. Dr. Chen is a member of the Institute of Electrical and Electronics Engineers. He has received three IBM Invention Achievement Awards and one IBM Outstanding Innovation Award for his work on error-correcting codes.

Donald T. Tang *IBM Research Division, P.O. Box 218, Yorktown Heights, New York 10598.* Dr. Tang is currently manager of VLSI projects on physical design algorithms in the Department of Computer Science. He received his B.S. from the National Taiwan University and his Ph.D. from the University of Illinois, both in electrical engineering. He joined IBM at the Thomas J. Watson Research Center in 1960 and has conducted research in various areas including network flows, error control coding, magnetic recording, network optimization, satellite switching, and VLSI testing and verification. He was on leave from the Research Center from 1965 to 1966 as a Visiting Associate Professor at the National Taiwan University and from 1978 to 1979 as a Program Manager at the IBM World Trade Americas/Far East Corporation. Dr. Tang is a Fellow of the Institute of Electrical and Electronics Engineers and a member of Eta Kappa Nu, Sigma Xi, and the International Scientific Radio Union.