Diagnosis by Signature Analysis of Test Responses

Mark G. Karpovsky, *Fellow, IEEE*, Lev B. Levitin, *Senior Member, IEEE*, and Feodor S. Vainstein

***Abstract*— We propose a new approach for identification of faulty processing elements based on an analysis of the compressed test response of the system. The test response is compressed first in space and then in time, and faulty processing elements are identified by hard decision decoding of the corresponding space-time signature. The approach results in considerable savings in hardware required for diagnostics.**

***Index Terms*— Diagnostic for computer systems and array processors, signature analysis, hard decision decoding, space-time compression of test responses.**

I. Diagnosis by Space-Time Compression of Test Responses

LET us consider the diagnosis problem for a system of (not necessarily identical) processing elements (e.g., a systolic array). The system is represented by a directed graph G whose nodes correspond to processing elements (PE's) and whose directed edges correspond to communication links. Our approach to the diagnosis problem is based on signature analysis of test responses. Signature analysis has been widely used for chip and board level testing and diagnosis [1]–[12].

The common approach to fault location is based on verification of signatures for each of the output PE's of the system, and analysis of the signature distortions. This approach results in a considerable overhead which grows proportionally to the number of output PE's in the system. Another approach which makes use of space compression of test responses before signature analysis was proposed in [11] and [12] for the case when only one PE in the system may be faulty, and this was generalized in [19] for the case of multiple faults. This approach results in a significant reduction of the required overhead, but it can be used only in the case when communications between PE's in the testing mode are implemented by the system bus or when each one of the PE's has provisions for a boundary scan [20].

The straightforward approach to diagnostics by signature analysis is illustrated by Fig. 1. Test responses $y(t) = (y_1(t), \cdots, y_n(t))$, at the moment t ($y_i(t)$ is a b-bit binary vector), are transferred via the system bus into a redundant chip in such a way that the response $y_i(t)$ at the output i is compressed in time by a b-bit linear feedback shift register with parallel load (LFSR i). After all test responses

Manuscript received March 7, 1991; revised February 4, 1992. This work was supported by the National Science Foundation under Grant MIP-8208437 and by NATO under Grant 910411.

M. G. Karpovsky and L. B. Levitin are with the Department of Electrical, Computers and Systems Engineering, College of Engineering, Boston University, Boston, MA 02215.

F. S. Vainstein is with the Department of Electrical Engineering, North Carolina A&T State University, Greensboro, NC 27411.

IEEE Log Number 9212690.

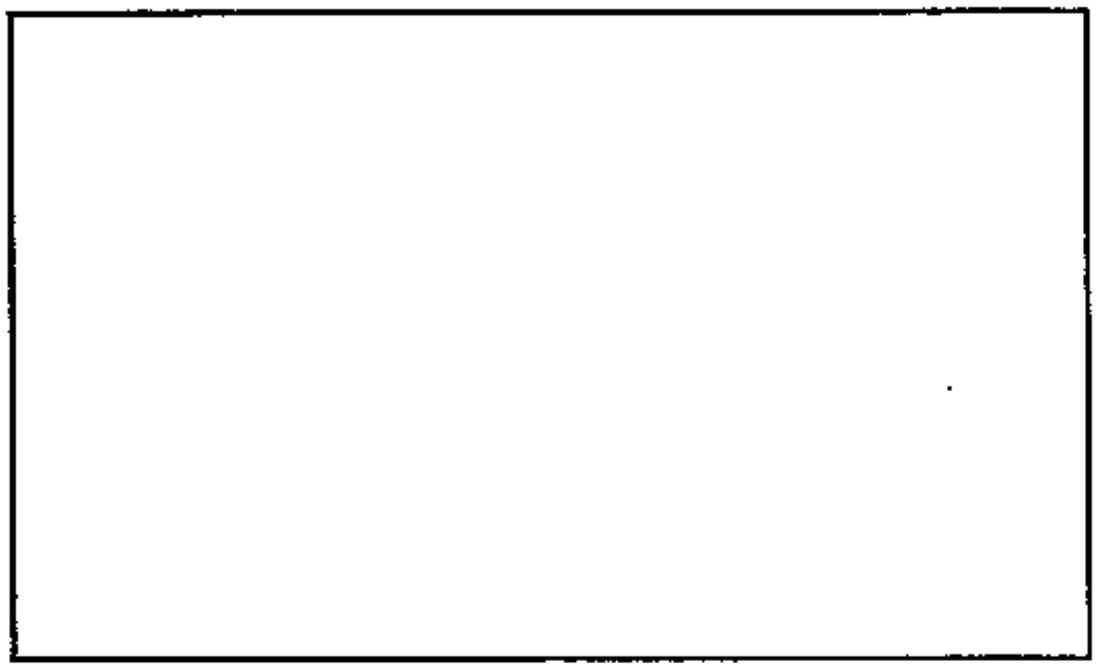


Fig. 1. The straightforward approach to diagnostics.

$y(1), \cdots, y(T)$ (T is the number of test patterns) have been compressed by the b-bit LFSR's with parallel inputs, the corresponding signatures $s_1, \cdots, s_n$ are compared to the precomputed reference signatures $s_1^0, \cdots, s_n^0$, and the error vector $e = (e_1, \cdots, e_n)$ is computed, where

$$e_i = \begin{cases} 1, & s_i \neq s_i^0; \\ 0, & s_i = s_i^0. \end{cases} \quad (1)$$

The identification of a faulty PE is implemented by an $(n \times N)$ decoder (N is the total number of PE's in the system) with the input $e = (e_1, \cdots, e_n)$.

We assume that a number of test patterns T is sufficiently large, so that a fault in a PE will manifest itself by distortions of signatures corresponding to all output PE's connected with the faulty PE.

For example, if the original array is the binary balanced tree (Fig. 2), a fault in PE_2 will result in error vector 11110000. (The time compression is performed by b-bit LFSR's with parallel load. We assume that the fault is not masked in any one of the eight LFSR's with parallel load. We assume that the fault is not masked in any one of the eight LFSR's compressing in time eight sequences $y_i(1), \cdots, y_i(T) (i = 1, \cdots, 8)$. The probability of masking is very small for large b [1], [7]–[9].) The relation between faulty PE's and error vectors for the binary tree of Fig. 2 is given in the first two columns of Table I.

In Sections I and II, we assume that at most one PE, or any number of incoming communication links to this PE, may be faulty and, as a result of these faults, the correct b-bit binary output of this PE is replaced by any other b-bit vector at any moment of time. The generalization of our approach for location of multiple faults is presented in Section III.

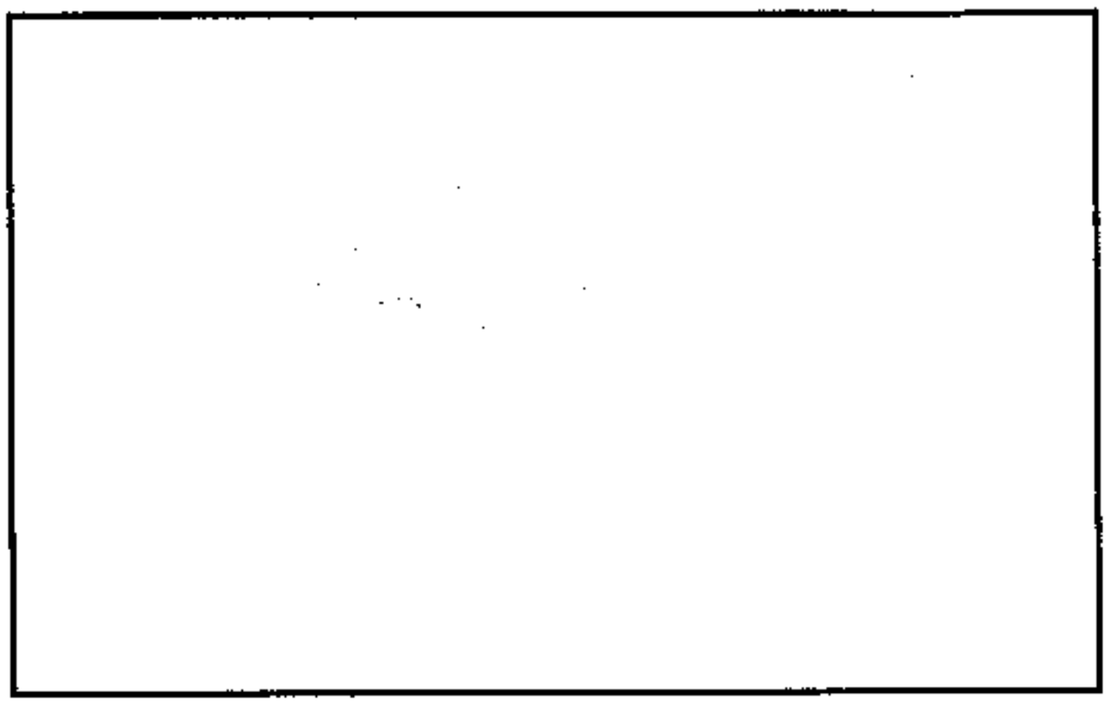
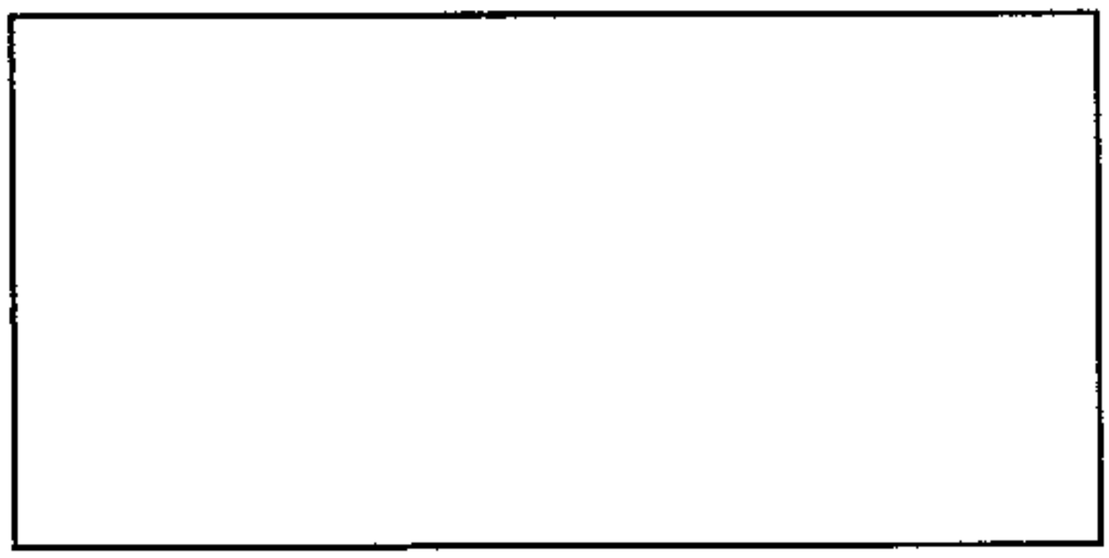


Fig. 2. Three-level balanced tree of PE's.

TABLE I
RELATIONS BETWEEN FAULTY PE'S, ERROR VECTORS, AND ERROR SYNDROMES FOR THE THREE-LEVEL BINARY TREE



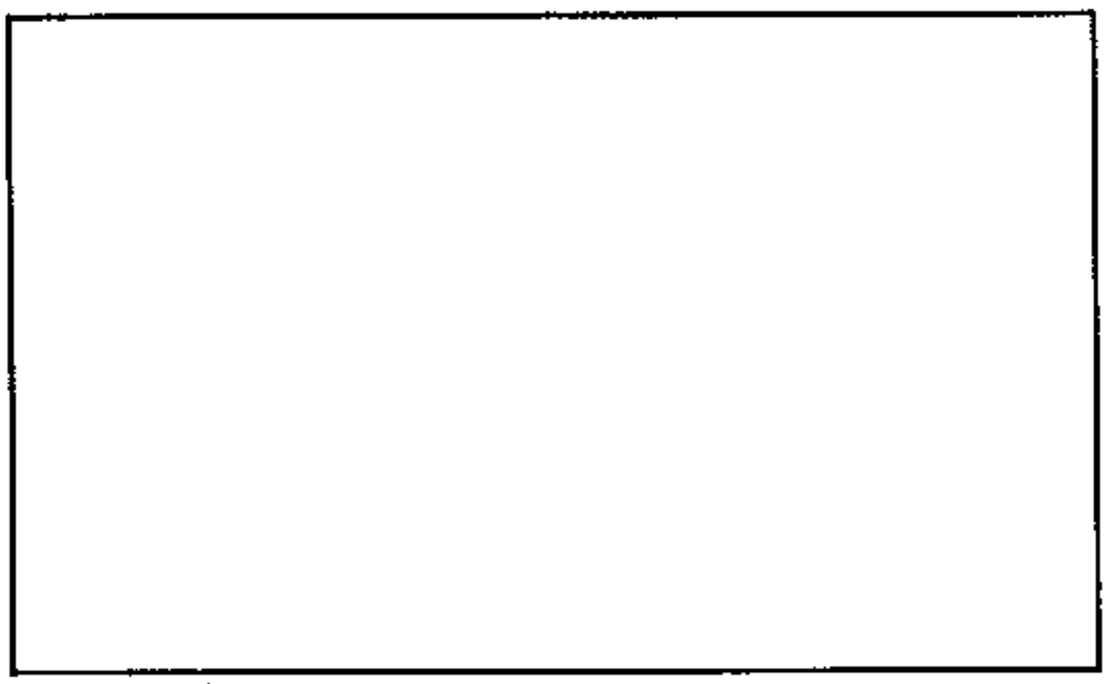


Fig. 3. An example of nondiagnosable systolic array.

The system is diagnosable iff all n-bit error vectors are different and not equal to $(0, \cdots, 0)$. Since the number of different error vectors is equal to the number N of PE's in the system, we have the following lower bound on a number of outputs n of a diagnosable system:

$$n \geq \lceil \log_2(N+1) \rceil. \tag{2}$$

Note that lower bound (2) is attainable (see, e.g., binary hypercube arrays in Section II). An example of a nondiagnosable system is given in Fig. 3. In this case, faults in PE_2 and PE_5 cannot be distinguished, since in both cases $e = (011)$.

Denote $PE_j < PE_i\,(G)$ if there exists a path from PE_i to PE_j in G. Then two systems G_1 and G_2 with the same set of PE's are equivalent from the diagnostic point of view when $PE_j < PE_i\,(G_1)$ iff $PE_j < PE_i\,(G_2)$. For example, arrays represented by Fig. 6(a) and (b) are equivalent.

For the straightforward approach to diagnostics represented in Fig. 1, the required hardware overhead L_1, in terms of a number of equivalent two-input gates, is of the order of $L_1 = O(bn) + O(Nn)$. For example, for the eight-level binary tree with $b = 32$, we have $n = 128, N = 255$, and $L_1 \simeq 110000$ (assuming that one flip-flop is equivalent to eight gates).

In this paper, another approach to diagnostics is described which results in a considerable reduction of the required overhead, while the probability of missing a fault remains small. It is shown below that, in many cases, the overhead can be decreased to $L_2 = O(b \log_2 n) + (N \log_2 n)$. This approach does not require redesigning and introducing self-test into PE's. Fault location, in this case, is implemented by an additional fault-free PE which generates test patterns, compresses test responses first in space and then in time, and analyzes the compressed responses (signatures) to identify faulty PE's. The structure of this additional PE does not depend on the structures of PE's in the original system.

The presented diagnostic approach may be used to provide for fault-tolerance in the system. In this case, the error locating procedure is implemented by a host PE which is assumed to be fault-free. After completing the diagnostic phase, the host initiates the reconfiguration phase to bypass the faulty PE. The algorithms for reconfiguration and their implementation have been developed, e.g., in [16] and [17].

To illustrate our approach to diagnostics, let us go back to the example of the three-level binary tree with $n = 8, N = 15$ (Fig. 2). Instead of compressing in time the sequence $y(1), \cdots, y(T)$ (where $y(t) = (y_1(t), \cdots, y_8(t))$ and $y_i(t)$ is a b-bit binary vector) by eight b-bit LFSR's with parallel load, we first compute $z(t) = Hy(t)$, where

$$H = \begin{bmatrix} 01000100 \\ 10100000 \\ 00010001 \\ 00001010 \\ 00100111 \end{bmatrix} \tag{3}$$

and all the computations are performed modulo two. This is the space compression step which results in $z(t) = (z_1(t), \cdots, z_5(t))$. Now, we will compress the sequence $z(1), \cdots, z(T)$ in time using only five b-bit LFSR's. The resulting five signatures $s_1, \cdots, s_5$ are compared to the precomputed five reference values $s_1^0, \cdots, s_5^0$, and the identification of a faulty PE is made by analyzing the error syndrome (the compressed error vector) $e^c = (e_1^c, \cdots, e_5^c)$, where $e_i^c = 1$ iff $s_i \neq s_i^0$ and $e_i^c = 0$, otherwise. For example, if PE_5 is faulty, then by (3), $e^c = (01101)$. Error syndromes e^c for different faults in this tree are presented in the rightmost column of Table I. Since different faults result in different nonzero syndromes e^c, identification of a faulty PE can be implemented by decoding e^c. Thus, we have been able to

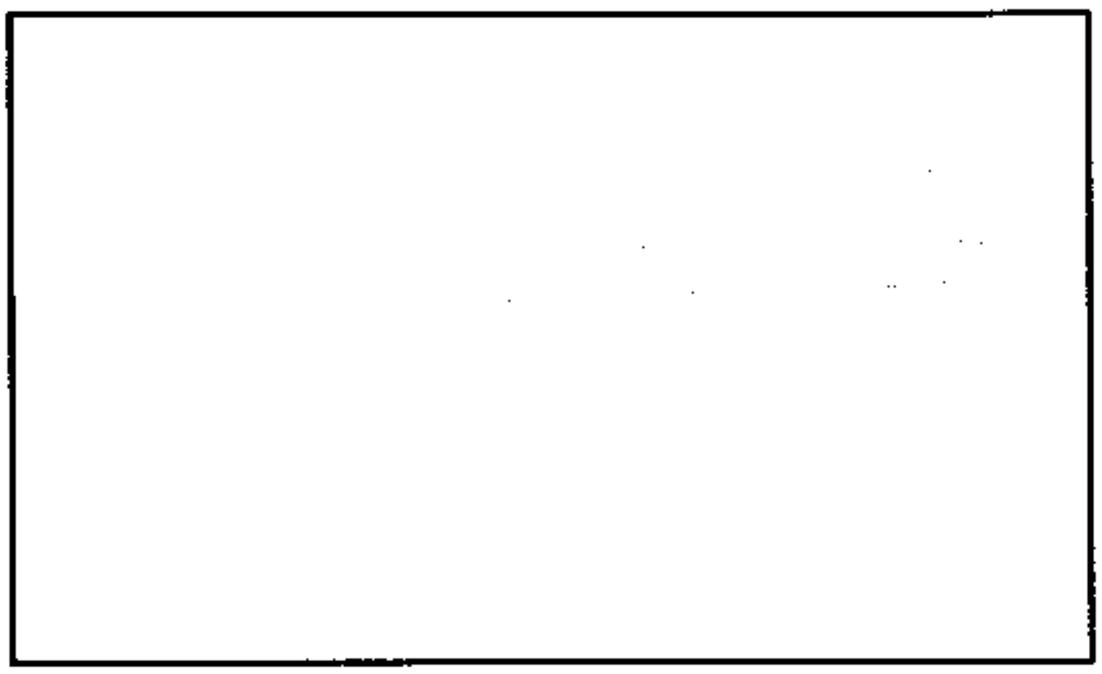


Fig. 4. Space-time approach for diagnostics.

reduce the overhead (using only five LFSR's and five reference values, instead of eight for the original approach), and we can still identify a faulty PE.

The block diagram for the proposed diagnostic approach based on the space-time compression of test responses is given in Fig. 4. The output response vector $y(t) = (y_1(t), \cdots, y_n(t))$ is compressed in space into $z(t) = (z_1(t), \cdots, z_r(t))$ where $y_i(t)$ and $z_j(t)$ are binary b-bit vectors, $z(t) = Hy(t)$ and H is a binary $(r \times n)$ matrix $(r \leq n)$, and multiplication of H by $y(t)$ is performed over $GF(2)$. This space compression is implemented by an H-counter modulo n and r T-flip-flop registers $T_1, \cdots, T_r$. The sequence of output vectors for the H-counter is the sequence $(h_1, \cdots, h_n)$ of r-bit columns of matrix H. Space signatures $z(t) = (z_1(t), \cdots, z_r(t))$ are compressed in time by r LFSR's with parallel load. Final space-time signatures $s_1, \cdots, s_r$ are compared to the precomputed reference values $s_1^0, \cdots, s_r^0$, and the resulting error syndrome $e^c = (e_1^c, \cdots, e_r^c)$ $(e_i^c = 1$ iff $s_i \neq s_i^0)$ is decoded to indicate the faulty PE. This identification is possible iff there is a one-to-one correspondence between PE's and error vectors $e^c = (e_1^c, \cdots, e_r^c)(e_i^c \in \{0, 1\})$. This mapping $e^c(i)(i = 1, \cdots, N)$ defines an embedding of the graph G representing the original system of PE's into the r-dimensional binary cube. The set of vertices of the r-dimensional binary cube (i.e., the set of all r-bit binary vectors) is a partially ordered set: we consider vector a to be a descendant of vector b, if a can be obtained from b by replacing some of the components equal to 1 by zeros. (It is said also that b covers a.) The embedding of graph G into the r-dimensional cube must preserve the partial ordering on G defined by its directed edges, i.e., if (i, j) and (i, q) are directed edges in G, then $e^c(i) = e^c(j) \vee e^c(q)$, where $\vee$ stands for the componentwise α operation. The embedding of the three-level binary tree into the five-dimensional binary cube is given by the rightmost column of Table I.

An overhead for the space-time compression (including the space compression hardware overhead) is of the order of $L_2 = O(br) + O(Nr)$. Comparing to the overhead L_1 for the straightforward approach, we have

$$\frac{L_1}{L_2} \simeq \frac{n}{r}. \tag{4}$$

Since $r \leq n$, the space-time compression technique is more efficient than the straightforward approach. To minimize the overhead, one has to minimize the length r of syndromes e^c.

Since all r-bit binary error syndromes e^c must be different and not equal to $(0, \cdots, 0)$ for all N PE's in the system, the following (attainable) bounds hold:

$$\lceil \log_2(N + 1) \rceil \leq r \leq n. \tag{5}$$

The overhead minimization problem for the space-time signature diagnostics can be reduced to constructing an $(r \times n)$ binary matrix H with minimal r such that the system remains diagnosable after the space compression $z(t) = Hy(t)$ of its output $y(t)$. This problem will be considered in Section II.

It is easy to show [15] that the relation between the error vectors e in the original system and error syndromes e^c is given by the following formula:

$$e^c = H \otimes e, \tag{6}$$

where $\otimes$ stands for the Boolean multiplication of an $(r \times n)$ binary matrix H by an n-bit binary vector e with addition being replaced by $\underline{\alpha}$. For example, for the binary tree of Fig. 2 with PE_5 being faulty, we have from (3): $e = (00110000)^{tr}$ and

$$e^c = \begin{bmatrix} 01000100 \\ 10100000 \\ 00010001 \\ 00001010 \\ 00100111 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \tag{7}$$

which corresponds to the fifth row in Table I.

Thus, the overhead minimization problem can be formulated in the following way: construct a space compression matrix H with a minimal number of rows such that for any two error vectors e and e'

$$H \otimes e \neq H \otimes e', \qquad H \otimes e \neq 0, \qquad H \otimes e' \neq 0. \tag{8}$$

The set of error vectors e is defined by the topology of interconnections in the original system, and the number of error vectors is equal to N.

It is remarkable that condition (8) on matrix H is similar to the necessary and sufficient condition for a check matrix of a code correcting error patterns defined by the graph G. The major difference is, however, that in our case operations over $GF(2)$ are replaced by the Boolean operations. The solution for the overhead minimization problem for several important classes of systems is given in the next section.

To conclude this section, we note that the proposed space-time signature approach to diagnostics is based on the "hard decision" decoding of signatures $s = (s_1, \cdots, s_r)$, when we identify a faulty PE by the binary vector e^c which indicates the distorted components in s. The magnitudes of distortions are not important for the hard decision procedure. An alternative procedure is a "soft decision" decoding of $s = (s_1, \cdots, s_r)$

for the space-time signature diagnosis. In this case, the identification of a faulty PE is based on the analysis of magnitudes of distortions in components of s. Soft decision techniques have been developed in [11] and [12] for board-level space-time signature diagnosis in case of single faults and in [19] for multiple faults. Similar techniques for space-time diagnosis of multiprocessor systems with single and multiple faults have been developed in [13]. In [11]–[13] and [19], the assumption has been made that components of the system are disconnected in the testing mode. In this paper, we will consider only hard decision space-time techniques, but we will not require the PE's be disconnected in the testing mode. We assume that components of the system under test are interconnected in the testing mode the same way as in the computing one.

II. Diagnosis of Array Processors

It is shown in the previous section that the problem of hardware minimization can be reduced to the design of an optimal space compression matrix H with a minimal number of rows r, satisfying (8).

Let us start with a lower bound for r. The set of all possible error patterns $\{e^{(k)}\}$ defined by (1) is a partially ordered set: $e^{(k_1)} \leq e^{(k_2)}$ iff $e^{(k_2)}$ covers $e^{(k_1)}$. Suppose we have a chain of error patterns

$$e^{(k_1)} \leq e^{(k_2)} \leq \cdots \leq e^{(k_D)}. \tag{9}$$

(In particular, if there is a path in G of length d (i.e., with d PE's in the path) from an input PE to an output PE, then the error patterns corresponding to PE's on that path form such a chain with $D = d$, and maximum d is called the depth of the system.) Then, by (6), the corresponding error syndromes $e^c(e^{(k_j)})$ also form a chain of length D of r-dimensional vectors with increasing weights (the weight of a binary vector is the number of components equal to 1). Since the zero vector $e^c = (0, \cdots, 0)$ is reserved for error-free condition, we obtain

$$r \geq D \geq d \tag{10}$$

where D is the maximum length of a chain (9) of error patterns, and d is the depth of the system. Lower bound (10) is attainable, for instance, in the case of 1-dim arrays [Fig. 5(a)] and of 2-dim near-neighbor meshes [Fig. 5(b)]. It is obvious also that if the same error pattern $e^{(k_1)}$ belongs to two or more chains with distinct error patterns of a maximum weight, then r should exceed D

$$r \geq D + 1. \tag{11}$$

The last situation occurs for rhombic arrays [Fig. 6(a)] and for triangular arrays (Fig. 10). In this paper, we consider systems with $D = d$.

Lower bounds (5), (10), and (11) can be improved if we have additional information about the topology of the original system. Let $N(h)$ be the number of paths of length at least h in the original system. Then, for embedding the system in the r-dimensional binary cube, all endpoints of paths of length at least h should be encoded by nonzero r-bit binary vectors of

(a)

(b)

Fig. 5. (a) Line of PE's. (b) $(h \times w)$ near-neighbor mesh of PE's.

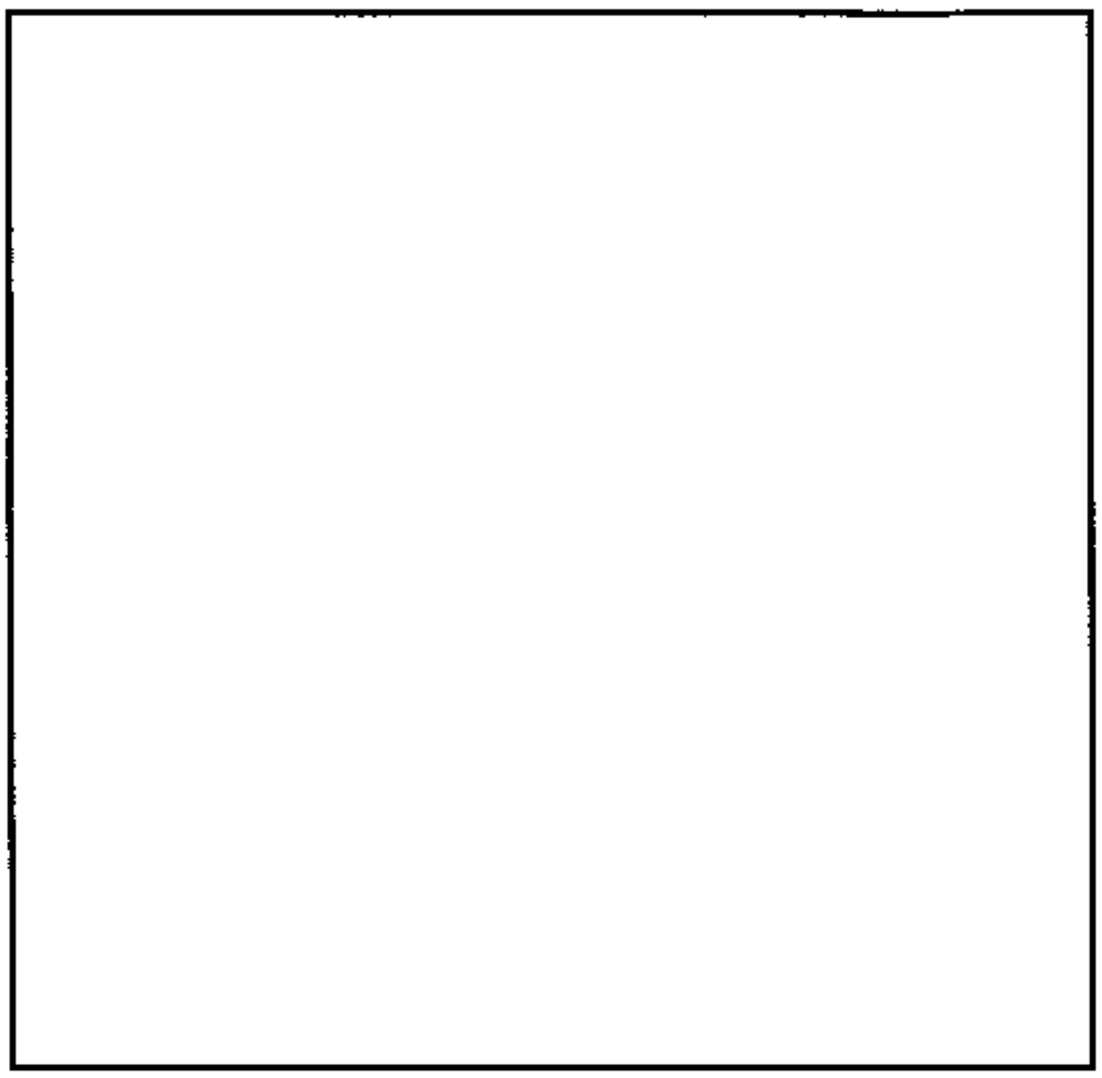
weight at most $r - h + 1$. Thus,

$$\sum_{i=1}^{r-h+1} \binom{r}{i} \geq N(h). \tag{12}$$

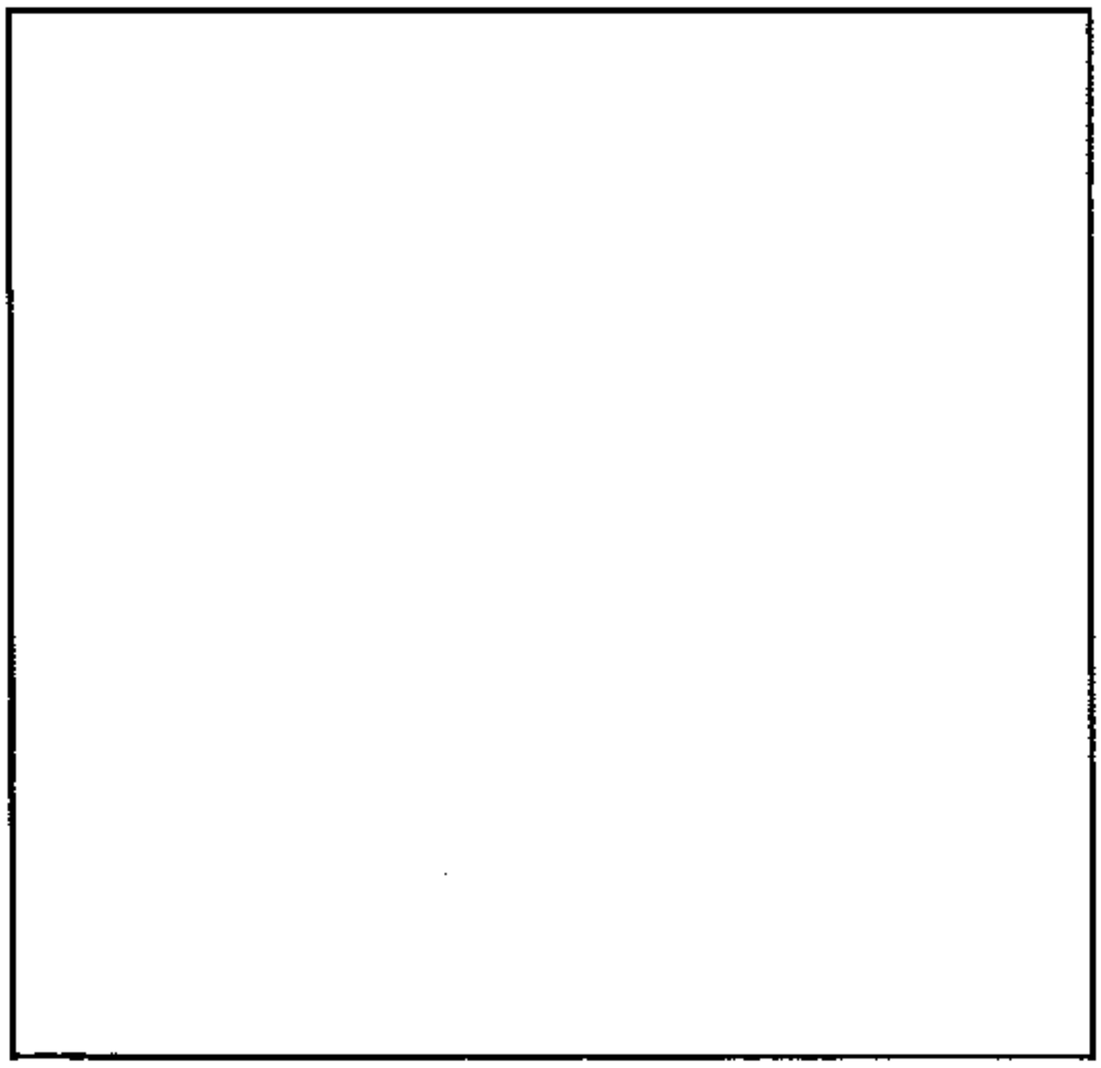
Another improvement of lower bounds (5) and (10) can be obtained for systems with many paths which do not have any endpoints in common [e.g., rhombic meshes, Fig. 6(a)]. Let $M(h)$ be a number of paths of length at least h which do not have any endpoints in common. If our array is embedded into a binary r-cube, then the binary r-dimensional vector corresponding to a node is a descendant of a vector corresponding to the previous node in the path. Thus, each path contains vectors of at least h different weights and the difference between the weights of endpoints of each path is at least $h - 1$. Consider now two "polar zones" in the r-cube: vectors of weights $w \geq \lceil \frac{r+h}{2} \rceil$ and of weights $w \leq \lceil \frac{r-h}{2} \rceil$. Then at least one of the endpoints for any path of length at least h belongs to one of the polar zones. Therefore,

$$\sum_{i=1}^{\lceil \frac{r-h}{2} \rceil} \binom{r}{i} + \sum_{i=\lceil \frac{r+h}{2} \rceil}^{r} \binom{r}{i} \geq M(h). \tag{13}$$

Lower bounds (12) and (13) are valid for all values of $h = 2, 3, \cdots, d$.



(a)



(b)

Fig. 6. (a) ($d \times n$) rhombic mesh of PE's. (b) Hexagonal mesh of PE's.

We will present below several nearly optimal constructions for space compression matrices H and lower bounds on minimal numbers of rows r in H for several important classes of systems: balanced trees (Fig. 2), 2-dim rhombic meshes [Fig. 6(a)], triangular meshes (Fig. 10), and cubic meshes (Fig. 11). These arrays have been widely used (see, e.g., [14] and [18]).

A. Diagnosis of Tree Processors

For the d-level binary tree T_d (d is the number of PE's on the path from the input to any output, $n = 2^{d-1}$, $N = 2^d - 1$), we

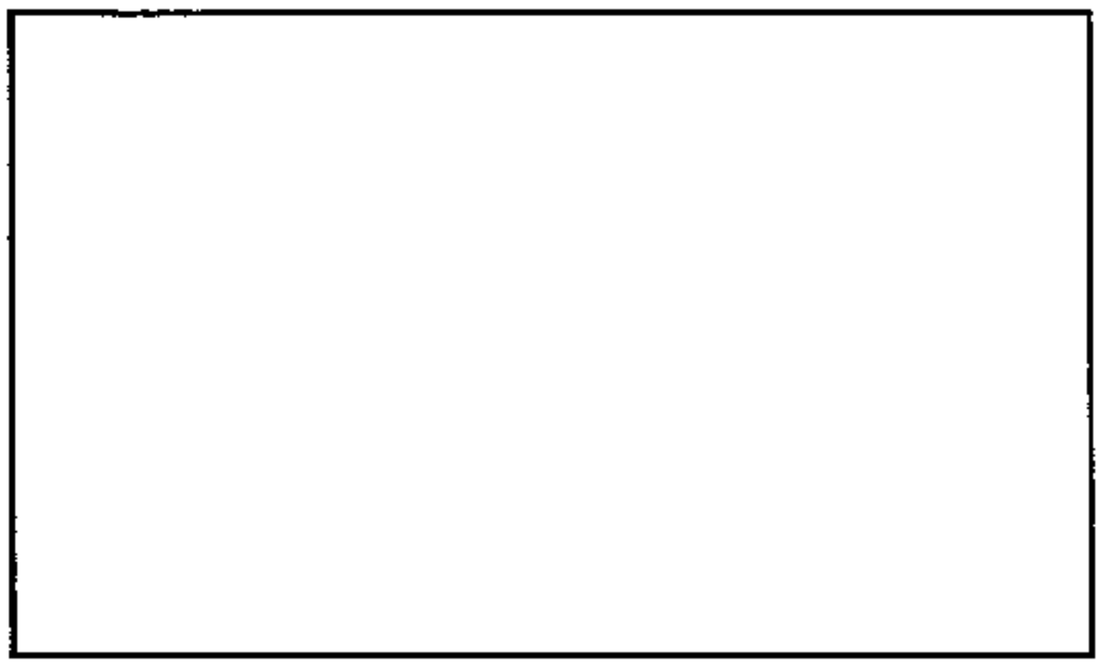


Fig. 7. Recursive construction for d-level binary tree T_d.

denote by $r(d)$ the minimal number of signatures to be stored, i.e., $r(d)$ is a minimal dimension of a binary cube $C_{r(d)}$ such that T_d can be embedded in $C_{r(d)}$ with preserving the partial ordering in T_d. For example, from Table I, we have $r(4) \leq 5$.

Since for the d-level binary tree $N(d) = 2^{d-1}$, we have from (12)

$$\sum_{i=1}^{r(d)-d+1} \binom{r(d)}{t} \geq 2^{d-1}. \tag{14}$$

Solving (14) for large d, we have

$$r(d) \gtrsim 1.29(d-1). \tag{15}$$

To construct an $r \times n$ space compression matrix H_d for T_d [which yields an embedding of the T_d into C_r and provides an upper bound for $r(d)$], we will use the recursive construction for the balanced binary tree T_d, shown in Fig. 7. Here $d = p + 1 - 1$ and $T_q^1, T_q^2, \cdots, T_q^P (P = 2^{p-1})$ are identical trees T_q of depth q with $Q = 2^{q-1}$ outputs.

Suppose that space compression matrices for T_p and T_q are $H_p = [h_p^1 h_p^2 \cdots h_p^P]$ and $H_q = [h_q^1 h_q^2 \cdots h_q^Q]$, respectively, where h_p^i and h_q^j are columns of H_p and H_q. Then it is readily seen that H_d can be constructed as follows:

$$H_d \left[\begin{array}{c|c|c|c} h_p^1 h_p^1 \cdots h_p^1 & h_p^2 h_p^2 \cdots h_p^2 & \cdots & h_p^P h_p^P \cdots h_p^P \\ h_q^1 h_q^2 \cdots h_q^Q & h_q^1 h_q^2 \cdots h_q^Q & \cdots & h_q^1 h_q^2 \cdots h_q^Q \end{array}\right]. \tag{16}$$

Thus, we have the following equation:

$$r(d) = r(p + 1 - 1) \leq r(p) + r(q). \tag{17}$$

An example of this construction for $d = 6, p = 4$, and $q = 3$ is shown in Fig. 9.

Matrix H_4 is given by (3), and the corresponding embedding of T_4 into C_5 is given by the rightmost column of Table I. From (3) and (14), we have $r(4) = 5$, which shows that the lower bound given by (12) is attainable.

Matrix H_5 and the corresponding embedding of T_5 into C_5 is given in Fig. 8, and H_6 is given in Fig. 9. By Fig. 8 and

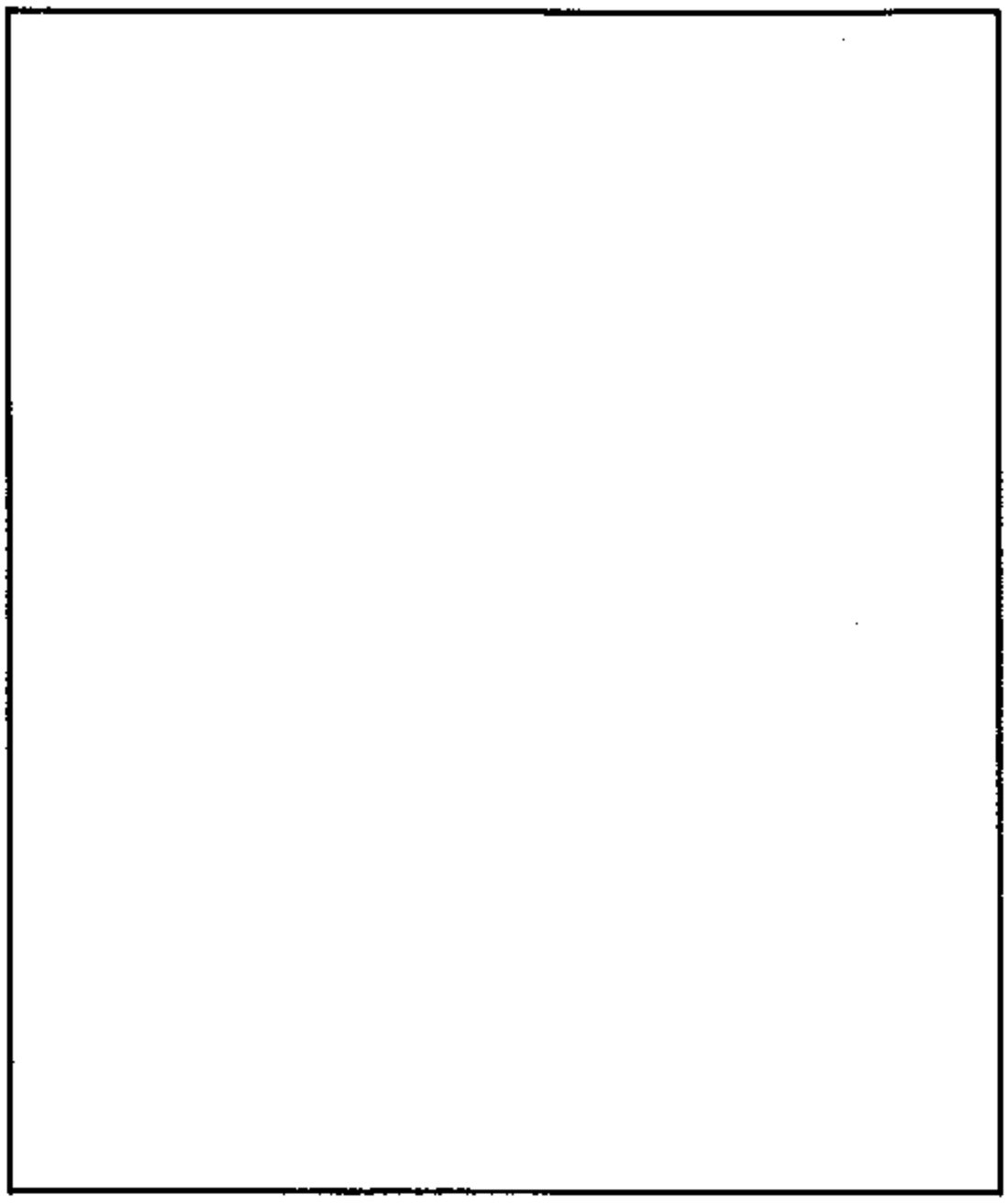


Fig. 8. Optimal space compression matrix H_5 and embedding of the five-level binary tree into 6-dimensional binary cube.

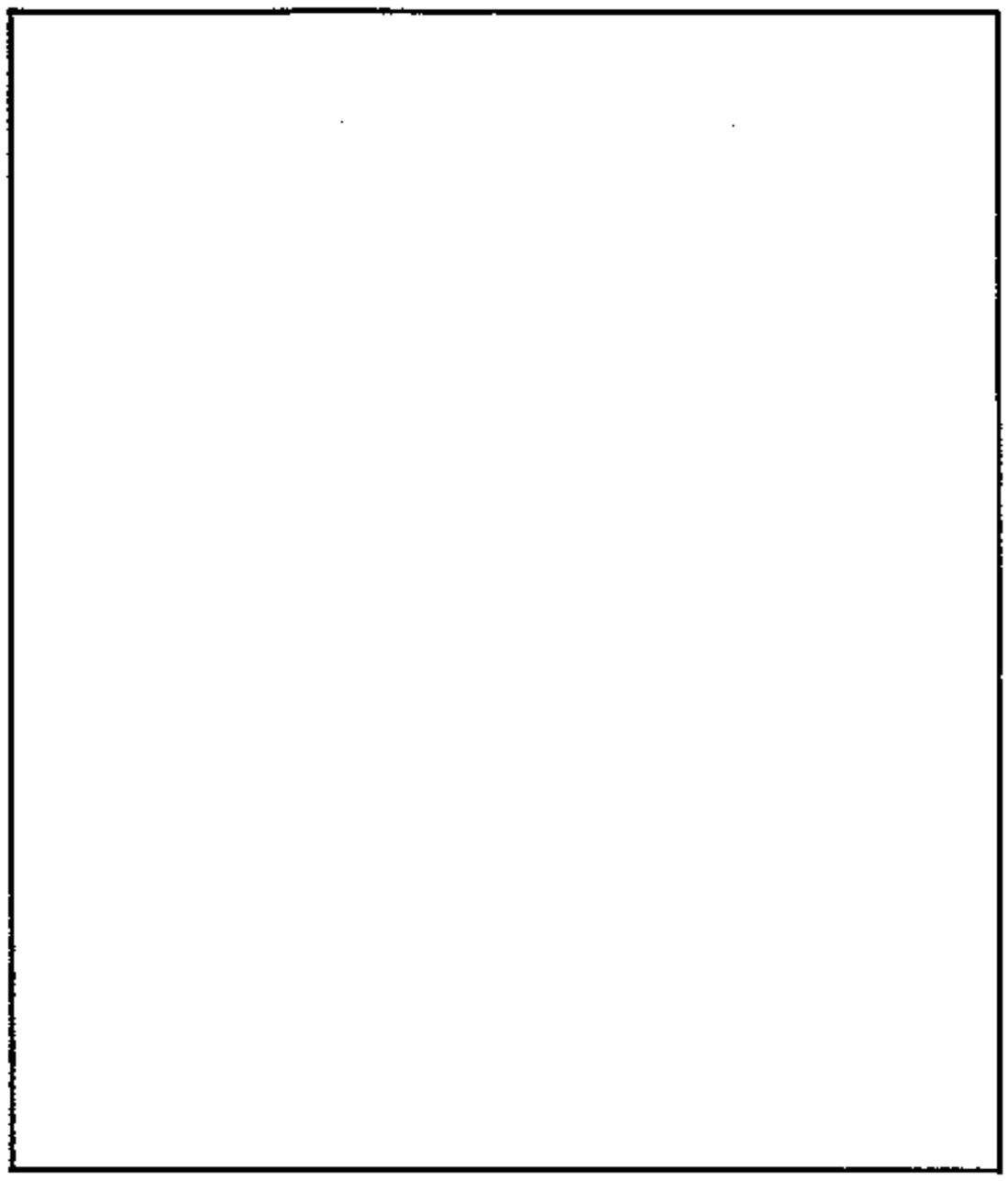


Fig. 9. Construction of the space compressions matrix for the binary trees T_3, T_4, and T_6 $(d = 3, 4, 6)$.

(14), we have $r(5) = 6$. Using this result and (17), we obtain

$$r(d) \le \begin{cases} \frac{3}{2}(d-1), & d = 4k+1; \\ \lceil \frac{3}{2}(d-1) \rceil + 1, & \text{otherwise}; \end{cases} \quad (18)$$

which is close to lower bounds (14) and (15). Some exact values of $r(d)$ and upper and lower bounds for $r(d)$ are given in Table II for $d \le 12$.

For the case of balanced p-ary trees ($p \ge 2$), lower bound (14) should be rewritten in the form

$$\sum_{i=1}^{r(d)-d+1} \binom{r(d)}{i} \ge p^{d-1}. \quad (19)$$

Asymptotically, for $d \gg 1$, it gives

$$\begin{aligned} r(d) &\ge 1.64(d-1) \quad \text{for } p = 3; \\ r(d) &\ge 2(d-1) \quad \text{for } p = 4; \\ r(d) &\ge (d-1)\log_2 p \quad \text{for } p \ge 5. \end{aligned} \quad (20)$$

Another lower bound for p-ary trees ($p \ge 2$) is given by (2)

$$r(d) \ge \left\lceil \log_2 \left(\frac{p^d - 1}{p-1} + 1 \right) \right\rceil. \quad (21)$$

For $p \ge 4$ and $d \gg 1$, (21) gives a tighter lower bound than (20).

Lower and upper bounds for $r(d)$ for different $p > 2$ are given in Table III for $N \le 5000$. The gap between bounds is very small.

TABLE II
MINIMAL NUMBERS OF SIGNATURES $r(d)$ REQUIRED FOR DIAGNOSTICS OF d-LEVEL BINARY TREES

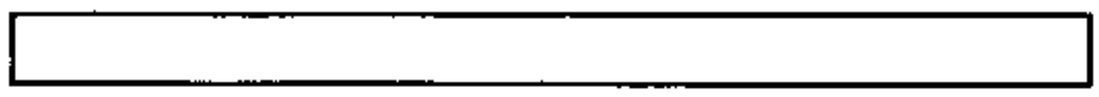
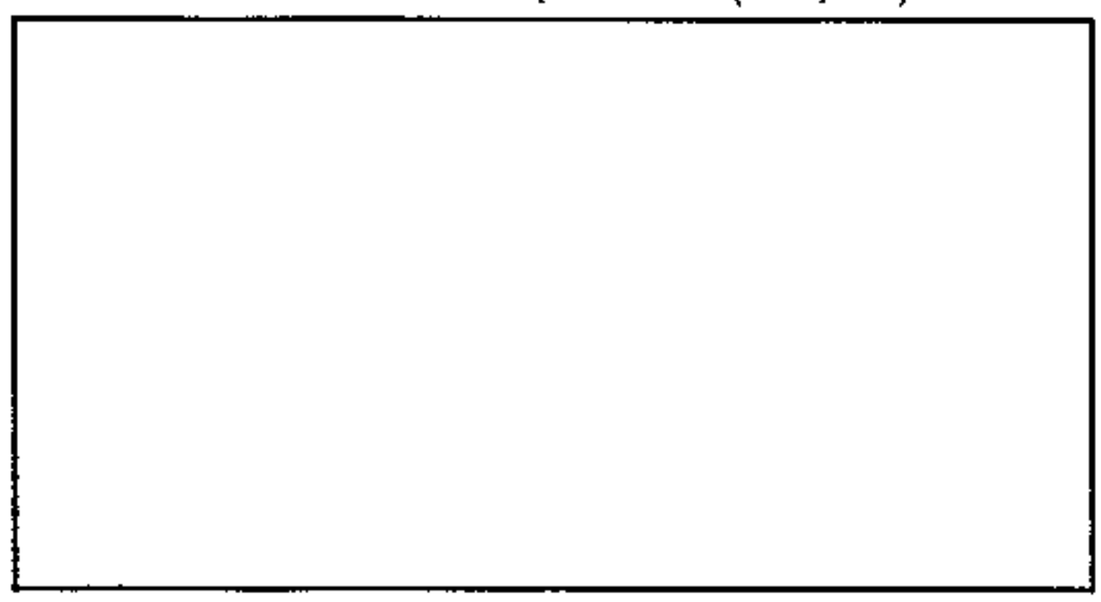


TABLE III
MINIMAL NUMBERS OF SIGNATURES $r_p(d)$ FOR DIAGNOSTIC OF d-LEVEL p-ARY TREES $(n = p^{d-1})$



Tables II and III illustrate considerable savings in hardware for the proposed space-time signature diagnostics approach over the straightforward diagnostic for binary trees. For example, for the binary tree with $N = 255$ processing elements $(d = 8, n = 128)$ and $b = 32$ output lines for every PE, assuming $r(8) = 11$ (see Table II), we have a reduction in hardware (measured in equivalent two-input gates) from $L_1 = 110\,000$ to $L_2 = 10\,000$.

B. Diagnosis of 2-dim Rhombic Arrays

The 2-dim rhombic mesh is shown in Fig. 6(a) (we assume that first and last columns in the mesh coincide). Denote by $r(n, d)$ the minimum number of rows in the H matrix that performs the space compression of test responses for this mesh. Obviously, the array is not diagnosable for $d \geq n$. The lower bounds for $r(n, d)$ following from (5) and (11) are

$$r(n, d) \geq \max\left(\lceil \log_2 (nd + 1) \rceil, d + 1\right). \tag{22}$$

A more specific lower bound can be obtained by the following reasoning. In a rhombic mesh of dimensions d and n, one can find n paths from the nodes of the top level to the nodes of the bottom level which do not have any endpoints in common. Thus, $M(d) = n$, and by (13) we have

$$n \leq \sum_{i=1}^{\lceil \frac{r-d}{2} \rceil} \binom{r}{i} + \sum_{i=\lceil \frac{r+d}{2} \rceil}^{r} \binom{r}{i}. \tag{23}$$

The minimum value of r that satisfies (23) is a lower bound for $r(n, d)$.

The construction of a matrix H for a rhombic mesh can be obtained in the following way. Consider two matrices H_1 and H_2, each of order $(d + 3) \times 3(d + 3)$ shown below:

$$H_1 = \left[\begin{array}{c|c|c} I_{d+1} & I_{d+1} & I_{d+1} \\ \hline 00\cdots0 & 00\cdots0 & 11\cdots1 \\ 00\cdots0 & 11\cdots1 & 00\cdots0 \end{array}\right]$$

$$H_2 = \left[\begin{array}{c|c|c} I_{d+1} & I_{d+1} & I_{d+1} \\ \hline 11\cdots1 & 00\cdots0 & 00\cdots0 \\ 00\cdots0 & 11\cdots1 & 00\cdots0 \end{array}\right] \tag{24}$$

where I_{d+1} is the $(d + 1)$-dimensional identity matrix.

Now let $n = 3(d + 1)m, k = \lceil \log_2 m \rceil$, where $m = 1, 2, \cdots$, and let g_l be the codeword for the integer $l (0 \leq l \leq 2^k - 1)$ in the k-bit Gray (reflexive) code. Denote by G_l a $k \times 3(d + 1)$ matrix which consists of identical columns g_l. Let B_l be the $(k + d + 3) \times 3(d + 1)$ matrix which is obtained by vertical concatenation (writing one matrix under the other) of matrices H_1 and G_l for an even l and H_2 and G_l for odd l. Then the space compression matrix H of order $r \times n$, where $r = k + d + 3$ and $n = 3(d + 1)m$, is obtained by the concatenation of matrices B_l in the order of increasing $l (l = 0, 2, \cdots, m - 1)$. Thus, we have (for m even) the equation given in (25) below.

An example of matrix H for $n = 18, d = 2 (k = 1, m = 2)$ is given at the bottom of this page, after (25).

$$H = [B_0 \mid B_1 \mid B_2 \mid \cdots \mid B_{m-1}] = \begin{bmatrix} H_1 & H_2 & H_1 & \cdots & H_2 \\ G_0 & G_1 & G_2 & \cdots & G_{m-1} \end{bmatrix}$$

$$= \left(\begin{array}{ccc|ccc|ccc|c|ccc} I_{d+1} & I_{d+1} & I_{d+1} & I_{d+1} & I_{d+1} & I_{d+1} & I_{d+1} & I_{d+1} & I_{d+1} & \cdots & I_{d+1} & I_{d+1} & I_{d+1} \\ \hline 0\cdots0 & 0\cdots0 & 1\cdots1 & 1\cdots1 & 0\cdots0 & 0\cdots0 & 0\cdots0 & 0\cdots0 & 1\cdots1 & \cdots & 1\cdots1 & 0\cdots0 & 0\cdots0 \\ 0\cdots0 & 1\cdots1 & 0\cdots0 & 0\cdots0 & 1\cdots1 & 0\cdots0 & 0\cdots0 & 1\cdots1 & 0\cdots0 & \cdots & 0\cdots0 & 1\cdots1 & 0\cdots0 \\ \hline g_0 & \cdots & g_0 & g_1 & \cdots & g_1 & g_2 & \cdots & g_2 & \cdots & g_{m-1} & \cdots & g_{m-1} \end{array}\right). \tag{25}$$

$$H = [B_0 \mid B_1] = \begin{bmatrix} H_1 & H_2 \\ G_0 & G_1 \end{bmatrix} = \left[\begin{array}{ccc|ccc} I_3 & I_3 & I_3 & I_3 & I_3 & I_3 \\ \hline 000 & 000 & 111 & 111 & 000 & 000 \\ 000 & 111 & 000 & 000 & 111 & 000 \\ \hline g_0g_0g_0 & g_0g_0g_0 & g_0g_0g_0 & g_1g_1g_1 & g_1g_1g_1 & g_1g_1g_1 \end{array}\right]$$

$$= \left[\begin{array}{c|c|c|c|c|c} 100 & 100 & 100 & 100 & 100 & 100 \\ 010 & 010 & 010 & 010 & 010 & 010 \\ 001 & 8001 & 8001 & 001 & 8001 & 8001 \\ \hline 000 & 000 & 111 & 111 & 000 & 000 \\ 000 & 111 & 000 & 000 & 111 & 000 \\ \hline 000 & 000 & 000 & 111 & 111 & 111 \end{array}\right].$$

TABLE IV
BOUNDS ON MINIMAL NUMBERS $r(n, d)$ OF SIGNATURES FOR RHOMBIC $(d \times n)$-MESHES

TABLE V
MINIMAL NUMBERS OF SIGNATURES $r(n, d)$ FOR RHOMBIC $(d \times n)$-MESHES WITH SMALL n AND d

It can be readily shown [15] that all the syndromes obtained by H designed above are different. The number of rows in H is given by

$$r = k + d + 3 = \left\lceil \log_2 \frac{n}{3(d+1)} \right\rceil + d + 3 \qquad (26)$$

and (26) provides an upper bound for $r(n, d)$.

Matrices H for $n \neq 3(d+1)m$ can be obtained by slight modifications of the construction given above.

The lower bounds given by (22) and (23) are attainable and sometimes coincide with the upper bound for $r(n, d)$ given by the above construction, which provides the exact value of $r(n, d)$.

In particular, for $n = 3(d+1)m$:

$$r(n, 2) = \lceil \log_2 m \rceil + 5 = \lceil \log_2 n - 2 \log_2 3 \rceil + 5 \qquad (27)$$

$$r(n, 3) = \lceil \log_2 m \rceil + 6 = \lceil \log_2 n - \log_2 3 \rceil + 4 \qquad (28)$$

$$\lceil \log_2 m \rceil + 6 \leq r(n, 4) \leq \lceil \log_2 m \rceil + 7 \qquad (29)$$

$$\lceil \log_2 m \rceil + 7 \leq r(n, 5) \leq \lceil \log_2 m \rceil + 8 \qquad (30)$$

$$d + 3 \leq r(6(d+1), d) \leq d + 4 \qquad (31)$$

$$r(3(d+1), d) = d + 3 \qquad (32)$$

$$r(d+2, d) = d + 2 \qquad (33)$$

$$r(d+1, d) = d + 1. \qquad (34)$$

The lower bound (L) based on (22) and (23), and the upper bound (U) based on (27)–(34) for $r(n, d)$, are presented in Table IV for $n \leq 2000$ and $d \leq 20$. Results for small n and d are shown in Table V. One can see from Tables IV and V that the gap between lower and upper bounds for $r(n, d)$ is very small.

Fig. 10. Triangular mesh of PE's.

Formulas (27)–(34) show that space-time signature diagnostic provides considerable hardware savings as compared to the straightforward approach (time compression only). For example, for the rhombic array with $n = 108, d = 8$, and $b = 32$, the straightforward approach requires approximately $L_1 \simeq 10^5$ equivalent two-input gates, while the suggested method requires only $L_2 \simeq 12 \times 10^3$ gates.

C. Diagnosis of Triangular Arrays

Let us now consider a problem of space compression for triangular meshes (see Fig. 10). (These meshes have been used for solution of dynamic programming problems [14].) For these meshes, $n = 2h - 1, N = 0.5h(h+1)$, and $d = h$. Since syndromes e^c should be different for faults in PE_1 and PE_{2h-1} (see Fig. 10), by (11), in this case,

$$r(d) \geq d + 1. \qquad (35)$$

The following construction for the space compression ma-

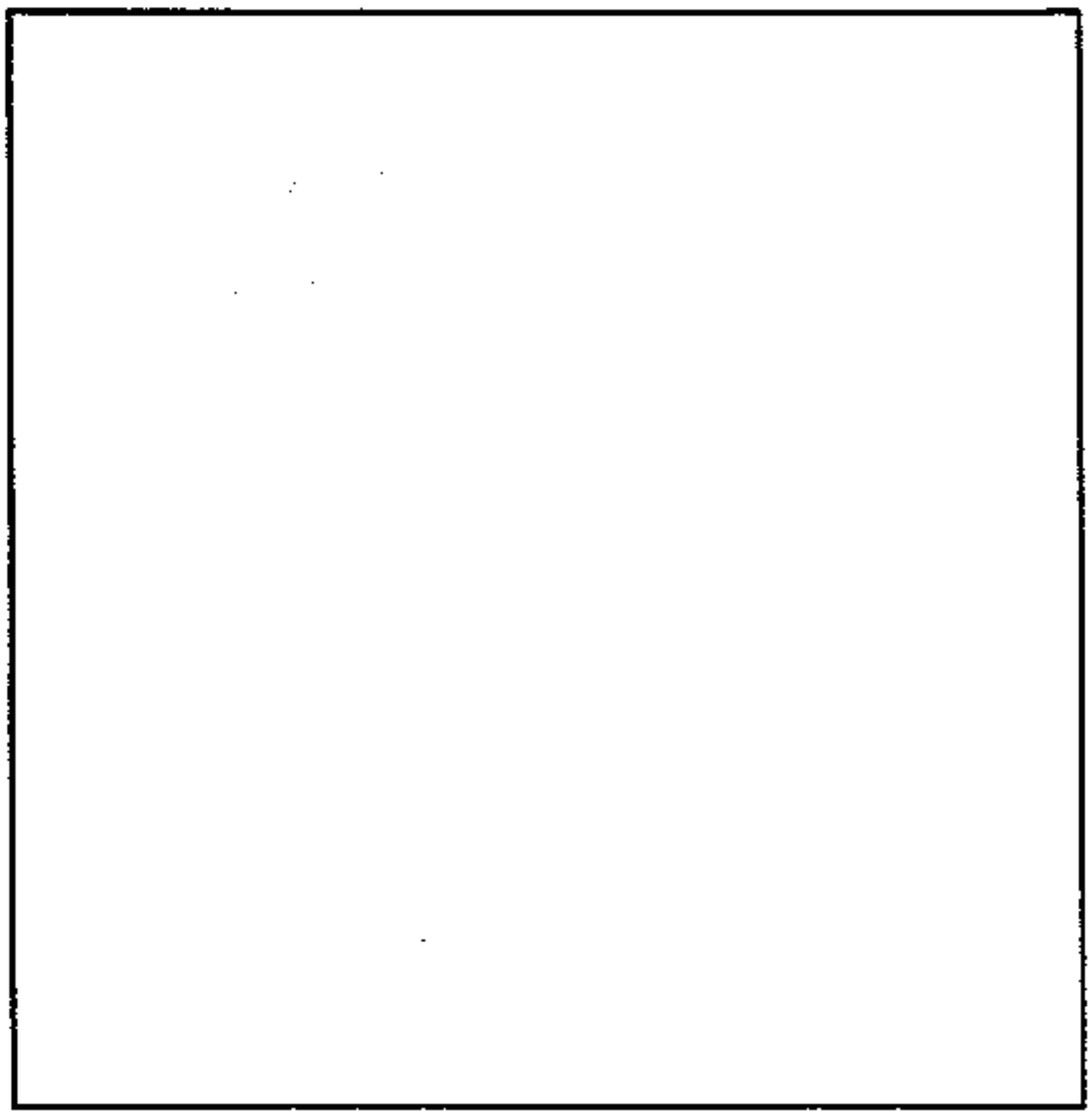


Fig. 11. 3-dim P-ary array of PE's.

trix H attains this lower bound:

$$H = \begin{bmatrix} 000\cdots01 & 1 & 00\cdots00 \\ 000\cdots10 & 0 & 00\cdots00 \\ 000\cdots00 & 0 & 00\cdots01 \\ 000\cdots00 & 0 & 00\cdots11 \\ 000\cdots00 & 0 & 00\cdots10 \\ & \cdots & \\ 010\cdots00 & 0 & 01\cdots00 \\ 100\cdots00 & 0 & 11\cdots00 \\ \underbrace{000\cdots00}_{d} & 1 & \underbrace{10\cdots00}_{d-2} \end{bmatrix} \Big\} d+1. \tag{36}$$

Thus,

$$r(d) = d+1, \qquad \frac{n}{r(d)} = \frac{2d-1}{d+1} \tag{37}$$

and about 50% reduction of an overhead can be obtained by the space-time signature diagnostics.

D. Diagnosis of Cubic Arrays

Let us now consider the problem of optimal space compression of outputs for 3-dim arrays (see Fig. 11). For the 3-dim $(p \times p \times p)$ array $(p \geq 2)$, a processor $PE_{i,j,k}$ with coordinates (i, j, k) have inputs from $PE_{i+1,j,k}$, $PE_{i,j+1,k}$, and $PE_{i,j,k+1}(i, j, k = 0, 1, \cdots, p-2)$. Boundary processors $PE_{p-1,j,k}$, $PE_{i,p-1,k}$, and $PE_{i,j,p-1}(i, j, k = 0, 1, \cdots, p-1)$ are inputs to the array, and processors $PE_{0,j,k}$, $PE_{i,0,k}$, $PE_{i,j,0}(i, j, k = 0, 1, \cdots, p-1)$ are outputs of the array. In this case, $N = p^3, n = 3p^2 - 3p + 1$, and $d = 3p - 2$. Thus, for this array,

$$r = r_3(p) \geq 3p - 2. \tag{38}$$

To compress the response from $n = 3p^2 - 3p + 1$ output PE's, one can monitor only $3p - 2$ PE's with coordinates $(i, 0, 0), (0, j, 0)$, and $(0, 0, k)(i, j, k = 0, 1, \cdots, p-1)$. Then, $PE_{I,J,K}$ is faulty iff I is a max i such that $e_{i,0,0} \neq 0, J$ is max j such that $e_{0,j,0} \neq 0$, and K is max k such that $e_{0,0,k} \neq 0$. Thus, for the p-ary 3-dim cubic array

$$r_3(p) = d = 3p - 2. \tag{39}$$

From (4),

$$\frac{L_1}{L_2} \simeq \frac{n}{r} = \frac{3p^2 - 3p + 1}{3p - 2} \simeq p \tag{40}$$

and for a large p, considerable savings in overhead can be obtained by the proposed space-time diagnostic.

Let us now generalize our previous results to the case of m-dimensional p-ary arrays $(m \geq 2, p \geq 2)$. For these arrays, every PE_i with coordinates $i = (i_1, i_2, \cdots, i_j, \cdots, i_m)$ and $i_j \in \{0, 1, \cdots, p-1\}$ have m inputs from PE's with coordinates $i = (i_1, i_2, \cdots, i_j + 1, \cdots, i_m)(j = 1, 2, \cdots, m)$. Boundary PE's with at least one of the coordinates being equal to $p-1$ are inputs to the array (there are $p^m - p^{m-1}$ such PE's), and PE's with at least one of the coordinates being equal to 0 are outputs (there are $n = p^m(p-1)^m$ output PE's). For these arrays,

$$N = p^m, \qquad \text{and } d = (p-1)m + 1. \tag{41}$$

To compress the responses from $n = p^m - (p-1)^m$ output PE's, one can monitor only $(p-1)m+1$ PE's with coordinates $(0, 0, \cdots, i, \cdots, 0), i \in \{0, 1, \cdots, p-1\}$. Then PE_I with $I = (I_1, I_2, \cdots, I_m)$ is faulty iff I_j is a max i_j such that $e_{0,0,\cdots,i_j,\cdots,0} \neq 0(j = 1, 2, \cdots, m)$. Thus, in a view of (41), we have for m-dim p-ary arrays

$$r = r_m(p) = (p-1)m + 1 \tag{42}$$

and

$$\frac{n}{r} = \frac{p^m - (p-1)^m}{(p-1)m+1} \simeq p^{m-2} \qquad \text{for large } p. \tag{43}$$

It follows from (42) that for $m = 2$, we have $n = r = 2p - 1$; but for $m > 2$, we have $r < n$, and savings from space-time diagnostics are increasing with an increase in a number of dimensions of a cubic array.

For the important case [14], [18] of binary hypercube arrays $(p = 2)$, we have

$$r = r_m(2) = m + 1 \tag{44}$$

which attains lower bound (2). In this case

$$\frac{n}{r} = \frac{2^m - 1}{m+1} \tag{45}$$

and considerable savings in overhead can be obtained by the space-time diagnostic. We note also that for binary cubes, coordinates $I = (I_1, I_2, \cdots, I_m)(I_j \in \{0, 1\})$ of a faulty PE can be computed as

$$I = (00\cdots01)e_{00\cdots01} \vee (00\cdots10)e_{00\cdots10} \vee (10\cdots00)e_{10\cdots00}.$$

In Table VI, we summarize our results on minimal numbers of signatures for different arrays (lines, 2-dim meshes, cubes,

TABLE VI
MINIMAL NUMBERS OF SIGNATURES AND COMPRESSION RATES FOR SPACE-TIME DIAGNOSTIC

binary and nonbinary balanced trees, 2-dim rhombic meshes, and triangular arrays).

III. LOCATION OF MULTIPLE FAULTS

In this section, we will apply the space-time diagnostic procedures developed in previous sections for location of multiple faults in arrays of PE's. To locate a fault with multiplicity l (l PE's in the system are faulty), we will use an l-step sequential procedure. At every step, we run the space-time diagnostic procedure described in Sections I and II, identify one faulty PE, replace it, and then repeat the procedure again. For example, for the three-level binary tree of PE's represented by Fig. 2 and a triple fault in PE_1, PE_6, and PE_{14}($l = 3$) by Table I at the first step, we identify that PE_1 is faulty, at the second step PE_6 and at the third step PE_{14}. For this example, $\eta_2 = 57.1\%$ of all double faults and $\eta_3 = 25.7\%$ of all triple faults will be located. Thus, using the same hardware required for location of single faults, one can locate a considerable portion of multiple faults by the multistep sequential error location.

We will estimate now fractions η_l of faults with multiplicity l which can be located by this procedure for each one of the arrays considered in Section II.

As was shown in Section II, the space-time diagnostic of array G results in an embedding of G into r-dimensional binary cube C_r where outputs of G are encoded by columns of the space compression matrix H. If PE_i, PE_j, and PE_k are encoded by $e^c(i), e^c(j)$, and $e^c(k)$, and there are links from PE_i to PE_j and from PE_i to PE_k, then $e^c(i) = e^c(j) \vee e^c(k)$.

A fault with multiplicity l in $PE_{i_1}, PE_{i_2}, \cdots, PE_{i_l}$ is locatable if and only if the set $\{e^c(i_1), e^c(i_2), \cdots, e^c(i_l)\}$ is totally ordered, i.e., for any $i_s, i_t \in \{i_1, i_2, \cdots, i_l\}$ either $e^c(i_s) \geq e^c(i_t)$ or $e^c(i_s) < e^c(i_t)$ ($e^c(i_s) = (e_1^c(i_s), e_2^c(i_s), \cdots, e_r^c(i_s)) \geq e^c(i_t) = (e_1^c(i_t), e_2^c(i_t), \cdots, e_r^c(i_t))$ iff $e_j^c(i_s) \geq e_j^c(i_t)$ for all $j = 1, 2, \cdots, r$). For example, in the tree of Fig. 2, the triple fault $\{PE_1, PE_6, PE_{14}\}$ is locatable since (see Table I) $e^c(i_1) = (11111) \geq e^c(i_6) = (10011) \geq e^c(i_{14}) = (00011)$ and $\{PE_1, PE_6, PE_{11}\}$ is not locatable since $e^c(i_6) = (10011)$ is not comparable with $e^c(i_{11}) = (00100)$.

We note that by the definition of the embedding $G \rightarrow C_r$, a fault in $\{PE_{i_1}, \cdots, PE_{i_l}\}$ is always locatable if there exists a path P in G such that $PE_{i_1}, \cdots, PE_{i_l} \in P$.

It was shown in the previous section that all single faults can be located by the space-time diagnosis. We will now estimate a fraction η_2 of locatable double faults for arrays considered in the previous sections.

Let N_2 be a number of pairs $\{PE_i, PE_j\}$ ($i \neq j, i, j = 1, \cdots, N$) such that $e^c(i) > e^c(j)$ or $e^c(j) > e^c(i)$. Then

$$\eta_2 = \binom{N}{2}^{-1} N_2. \tag{46}$$

We will apply formula (46) to estimate η_2 for systems with different topologies described in Sections I and II.

For 1-dim array [Fig. 5(a)], all faults with any multiplicity are locatable. For the 2-dim ($h \times w$)-near neighbor mesh [Fig.

5(b)], we have

$$\eta_2 = \binom{hw}{2}^{-1} \sum_{i=1}^{h}\sum_{j=1}^{w}(ij-1)$$
$$= \binom{hw}{2}^{-1}\left(\binom{h+1}{2}\binom{w+1}{2} - hw\right)$$
$$= 0.5\left(1+\frac{h+w-2}{hw-1}\right). \quad (47)$$

For the binary m-cube of PE's

$$\eta_2 = \binom{2^m}{2}^{-1}\sum_{i=0}^{m}(2^i-1)$$
$$= 2^{-m+1}(2^m-1)^{-1}(3^m-2^m) \quad (48)$$

and η_2 is decreasing exponentially with an increase in a number of dimensions.

For a p-ary 3-cube of processors

$$\eta_2 = \binom{p^3}{2}^{-1} \sum_{i_1,i_2,i_3=1}^{p}(i_1i_2i_3-1)$$
$$= \binom{p^3}{2}^{-1}\left(\binom{p+1}{2}^3 - p^3\right)$$
$$= 0.25\left(1+\frac{3p^2+3p-6}{p^3-1}\right) \quad (49)$$

and η_2 is converging to 0.25 with an increase in p.

For the general case of p-ary m-cube array of processors ($p \geq 2, m \geq 1$), we have

$$\eta_2 = \binom{p^m}{2}^{-1} \sum_{i_1,i_2,\cdots,i_m=1}^{p}(i_1i_2,\cdots,i_m-1)$$
$$= \binom{p^m}{2}^{-1}\left(\binom{p+1}{2}^m - p^m\right). \quad (50)$$

Formula (50) generalizes our previous results for the 1-dim array ($m=1$), 2-dim meshes (for $h=w=p$), binary cube ($p=2$), and p-ary 3-cube.

For a p-ary star array (p-ary tree with $d=2$ levels) with $p=2^i-2$, we have $r=i$ and

$$\eta_2 = \binom{2^r-1}{2}^{-1}\sum_{j=1}^{r}\binom{r}{j}(2^j-2)$$
$$= \binom{2^r-1}{2}^{-1}(3^r-2^{r+1}+1)$$
$$= \binom{p+1}{2}^{-1}((p+2)^{\log_2 3}-2p-3). \quad (51)$$

Thus, for p-ary stars, a fraction of locatable double faults is slowly decreasing with an increase in p.

TABLE VII
FRACTIONS OF DOUBLE η_2 AND TRIPLE η_3 FAULTS LOCATABLE BY SPACE-TIME DIAGNOSTIC IN A d-LEVEL BALANCED BINARY TREES

For the d-level binary balanced tree, we can construct a lower bound on η_2 by estimating the number $N_2 = N_2(d)$ of pairs $\{PE_i, PE_j\}$ of PE's such that PE_i and PE_j belong to the same path in the tree. In this case,

$$N_2(d) = \sum_{i=0}^{d-2}2^i(2^{d-i}-2) = (d-2)2^d+2 \quad (52)$$

and

$$\eta_2 \geq \binom{N}{2}^{-1} N_2(d) = \binom{2^d-1}{2}^{-1}((d-2)2^d+2). \quad (53)$$

These formulas can be generalized to the case of p-ary d-level trees. For p-ary trees,

$$N_p(d) = \sum_{i=0}^{d-2}p^i\left(\frac{p^{d-i}-1}{p-1}-1\right)$$
$$= (p-1)^{-1}\left((d-1)p^d - \frac{p^d-1}{p-1}+1\right) \quad (54)$$

and

$$\eta_2 \geq \binom{\frac{p^d-1}{p-1}}{2}^{-1}\frac{1}{p-1}\left((d-1)p^d-\frac{p^d-1}{p-1}+1\right). \quad (55)$$

We note that lower bounds (53) and (55) are rather weak for large d. Exact values of $\eta_2 = \eta_2(d)$ for binary trees are given in Table VII for $d \leq 8$.

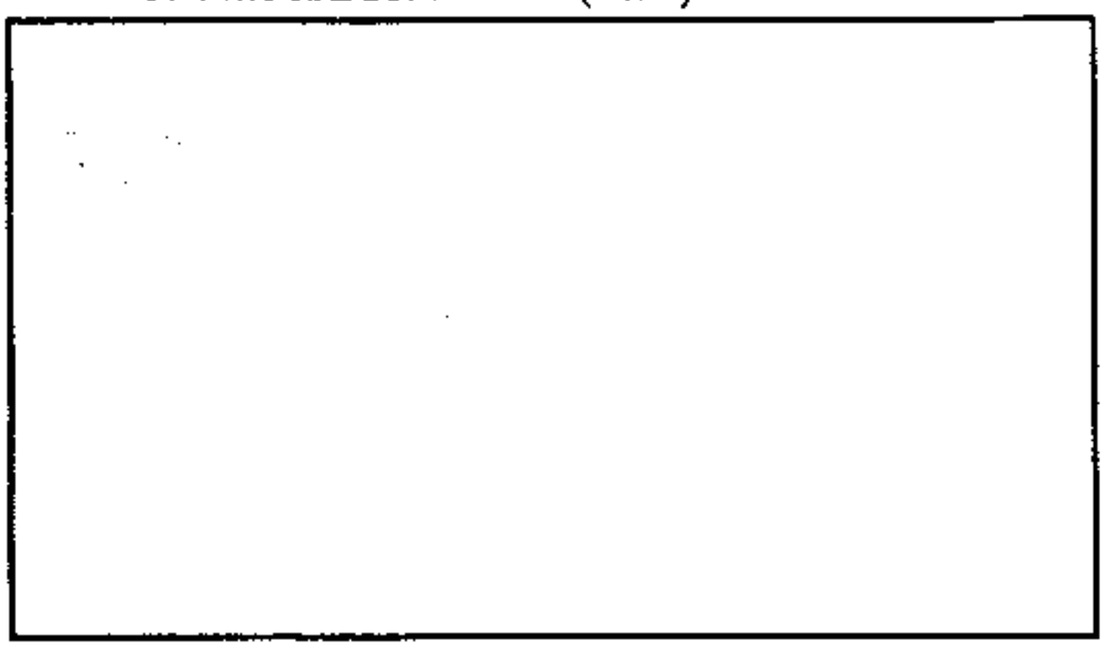
For the 2-dim ($d \times n$) rhombic meshes [Fig. 6(a)], the number $N_2(d, n)$ of pairs $\{PE_i, PE_j\}$, such that PE_i and PE_j belong to the same path, is

$$N_2(d,n) = n\sum_{i=0}^{d-2}\sum_{j=2}^{d-i}j$$
$$= \left(\binom{d+1}{2}-1\right)dn - n\sum_{j=2}^{d}j(j-1)$$
$$= \left(\binom{d+1}{2}-1\right)dn - \frac{1}{6}dn(d+1)(2d+1)$$
$$+ n\binom{d+1}{2} \quad (56)$$

and

$$\eta_2 \geq \binom{dn}{2}^{-1} N_2(d,n) = \frac{d^2+3d-4}{3(dn-1)}. \quad (57)$$

TABLE VIII
FRACTIONS OF DOUBLE η_2 AND TRIPLE η_3 FAULTS LOCATABLE BY SPACE-TIME DIAGNOSTIC IN $(d \times n)$-RHOMBIC MESHES



This lower bound is weak for $h \ll w$. Exact values of $\eta_2 = \eta_2(d, n)$ for $d \leq 10$ and $n \leq 33$ are given in Table VIII.

The last array we are going to consider in this section is the triangular p-ary (Fig. 10). For these arrays, we have for the number $N(p)$ of pairs $\{PE_i, PE_j\}$ such that PE_i and PE_j belong to the same path

$$N(p) = N(p-1) + \sum_{i=1}^{p}(i(p-i+1)-1)$$

$$= N(p-1) + p\binom{p+1}{2} - p - \sum_{i=1}^{p} i(i-1)$$

$$= N(p-1) + \binom{p+2}{2} - p. \tag{58}$$

Solving (58) with $N(1) = 0$, we have

$$N(p) = \frac{p(p^2-1)(p+6)}{24} \tag{59}$$

and

$$\eta_2 = \binom{p+1}{2}^{-1} N(p) = \frac{(p+6)(p-1)}{3(p(p+1)-2)}. \tag{60}$$

It follows from (60) that for large $p, \eta_2 \simeq 0.33$ for triangular p-ary arrays.

These results can be generalized for faults with multiplicity $l \geq 2$. For most arrays, the fraction η_l of locatable faults with multiplicity l is decreasing exponentially with increase in l. This fraction is close to one for hierarchical structures with $d/N \simeq 1$ (for example, for 1-dim array $\eta_l = 1$ for any l) and close to zero if $d/N \simeq 0$. The fractions η_3 of locatable triple faults for binary trees are presented in Table VII for $d \leq 8$, and in Table VIII for rhombic meshes with $d \leq 10, n \leq 33$.

One can see from the results presented in this section that for small l for many arrays, considerable fractions of faults with multiplicity $l > 1$ can be located by l-step sequential space-time diagnostic procedures.

IV. CONCLUSIONS

We presented a new method for identification of faulty processing elements. The method is based on compression of a test response first in space and then in time using multiple input LFSR's and hard decision decoding techniques. The overhead analysis and the solution for the hardware minimization problem are presented for several important classes of systems. The proposed method results in considerable hardware savings.

ACKNOWLEDGMENT

The authors are grateful to S. Chaudhry of Boston University for his valuable assistance in conducting computer experiments.

REFERENCES

[1] J. E. Smith, "Measures of the effectiveness of fault signature analysis," *IEEE Trans. Comput.*, vol. C-29, pp. 510–516, 1980.
[2] W. H. McAnney and J. Savir, "There is information in faulty signatures," *Proc. Int. Test Conf.*, 1987, pp. 630–636.
[3] J. Savir and W. H. McAnney, "Identification of failing test with cyclic registers," *Proc. Int. Test Conf.*, 1988, pp. 322–328.
[4] B. Koenemann, J. Mucha, and G. Zwiehoff, "Buitl-in test for complex digital integrated circuits," *IEEE J. Solid-State Circuits*, vol. SC-15, pp. 315–318, June 1980.
[5] S. J. Upadhyaya and K. K. Saluja, "Signature techniques in fault detection," in *Spectral Techniques and Fault Detection*, M. Karpovsky, Ed. New York: Academic, 1985, pp. 421–477.
[6] P. H. Bardell and W. H. McAnney, "Self-testing of multichip logic modules," in *Proc. Int. Test Conf.*, 1982, pp. 200–204.
[7] T. W. Williams, W. Daehn, M. Gruetzner, and C. W. Starke, "Bounds and analysis of aliasing errors in linear feedback shift registers," *IEEE Trans. Comput.-Aided Des.*, vol. 7, pp. 75–83, Jan. 1988.
[8] A. Ivanov and V. K. Agarwal, "An analysis of the probabilistic behavior of linear feedback shift registers," *IEEE Trans. Comput.-Aided Des.*, vol. 8, pp. 1074–1088, Oct. 1989.
[9] D. K. Pradhan, S. K. Gupta, and M. G. Karpovsky, "Aliasing probability for multiple input signature analyzer," *IEEE Trans. Comput.*, Apr. 1980.
[10] S. R. Reddy, K. K. Saluja, and M. G. Karpovsky, "A data compression for built-in self test," *IEEE Trans. Comput.*, vol. C-37, pp. 1151–1156, Sept. 1988.
[11] M. G. Karpovsky and P. Nagvajara, "Board level diagnosis," *Proc. Int. Test Conf.*, 1988, pp. 47–53.
[12] ——, "Design of self-diagnostic boards by signature analysis," *IEEE Trans. Ind. Eng.*, Apr. 1989.
[13] M. G. Karpovsky, "An approach for error detection and error correction in distributed systems computing numerical functions," *IEEE Trans. Comput.*, vol. C-30, pp. 947–954, Dec. 1981.
[14] K. Hwang and F. A. Briggs, *Computer Architecture and Parallel Processing*. New York: McGraw-Hill, 1984.
[15] M. G. Karpovsky, L. B. Levitin, F. S. Vainstein, "Identification of faulty processing elements by space-time compression of test responses," *Proc. Int. Test Conf.*, 1990.
[16] I. A. Koren, "A reconfigurable and fault-tolerant VLSI multiprocessor array," in *Proc. 8th Annu. Symp. Comput. Arch.*, 1981.
[17] I. A. Koren and I. Pomerantz, "Distributed structuring of processor arrays in the presence of faulty processors," in *Systolic Arrays*. New York: Adam Higler, 1986.
[18] S. Y. Kung, "VLSI array processors," in *Systolic Arrays*. New York: Adam Higler, 1987.
[19] M. G. Karpovsky and S. M. Chaudhry, "Built-in self-diagnostic by space-time compression," in *Proc. IEEE VLSI Test Symp.*, 1992.
[20] D. K. Pradhan, "Developing a standard for boundary-scan implementations," in *Proc. IEEE Int. Conf. Comput. Des.*, 1987, pp. 462–466.

Mark G. Karpovsky (F'[illegible]) is a Professor of computer engineering at Boston University and the Director of the Research Laboratory for Design and Testing of Computer Systems. His research interests include testing, fault-tolerant computing, design for testability, diagnostics of computer hardware, and error correcting codes. He has published over 100 papers and several books in these areas. He has also been a consultant for several companies, including IBM, Digital, Honeywell, and AT&T.

Lev B. Levitin (SM'[illegible]) received the M.Sc. degree in physics from Moscow University in 1960 and the Ph.D. degree in physical and mathematical sciences from the U.S.S.R. Academy of Sciences and Gorky University in 1969.

He worked at the Institute of Information Transmission Problems, the U.S.S.R. Academy of Sciences, from 1961 to 1973, and taught at Tel-Aviv University from 1974 to 1980. In 1980–1982 he was a Visiting Scientist with the Heinrich-Hertz-Institute, Berlin, DFVLR (German Space Research Center) in Oberpfaffenhoffen, Germany; and a Visiting Professor at Bielefed University, Germany; and Syracuse University, Syracuse, NY. He was also a Visiting Scientist at Regensburg University, Germany, and at Standard Electric Lorenz AG Research Center, Stuttgart, Germany. Since 1982 he has been with the College of Engineering, Boston University; and since 1986 he has been Distinguished Professor of Engineering Science with the Department of Electrical, Computer and Systems Engineering at Boston University. He has published about 80 papers and presentations. His research areas include physical information theory with applications to quantum communication systems; physics of computation; applications of information theory to statistical physics, quantum theory of measurements, linguistics, and theory of complex systems; coding theory; and theory of computer hardware testing.

Feodor S. Vainstein received the M.S. degree in electronics and electrical engineering and the M.S. degree in applied mathematics from Moscow Institute of Electrical and Computer Engineering, and the Ph.D degree in electrical engineering from Boston University in 1971, 1974, and 1992, respectively. He is currently an Associate Professor in the Department of Electrical Engineering at North Carolina A&T State University. He has taught at a number of schools in the U.S.S.R. From 1988 to 1992 he taught at Boston University in the Department of Electrical Engineering and in the Department of Computer Science. His interests include fault-tolerant computing, signal processing, and applied mathematics.

[redacted]

77%

Karpovsky x6

x7'2

C-Oct 93
Karpovsky etc

,16
1.53

Levitin, Lev B.

Vainstein, Feodor S

60%

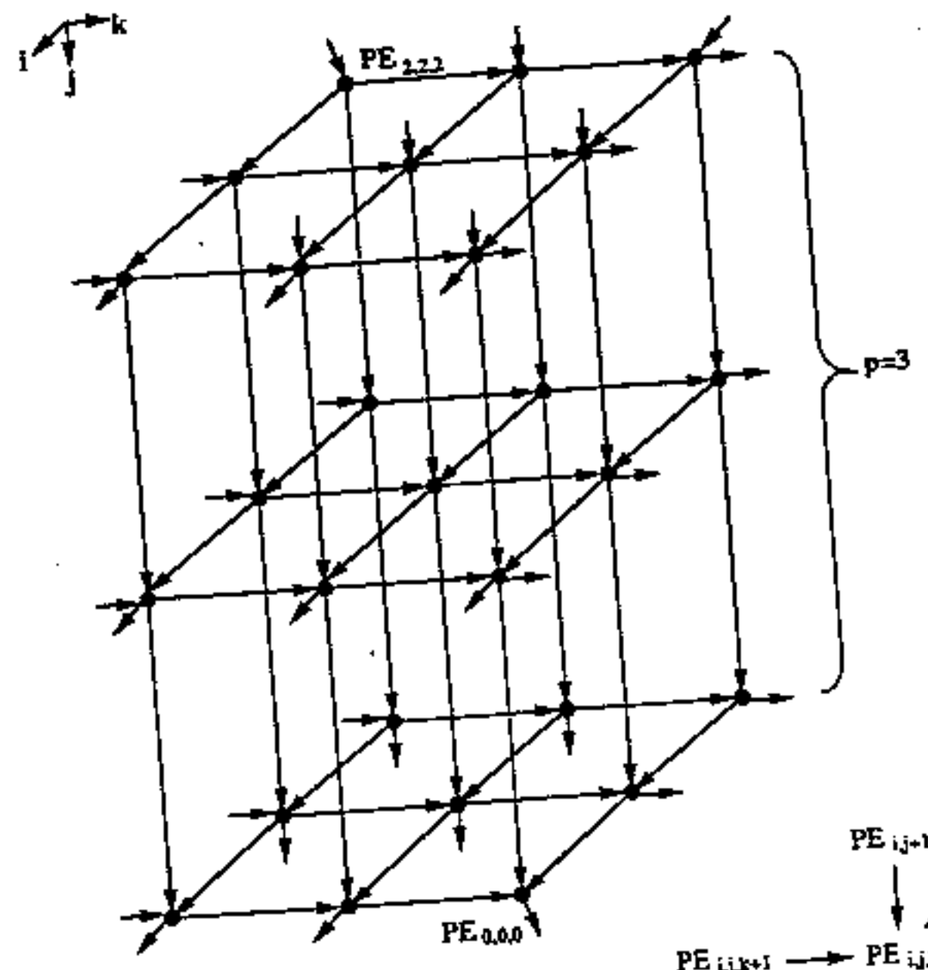
Vainstein x6

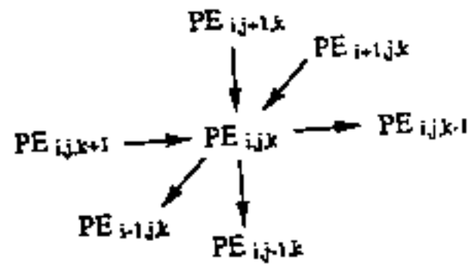
x 7½

C-Oct 53
Karpovsky et al

.09
1.27

48% (4t x21)





$n=3p^2-3p+1$, $N=p^3$, $d=3p-2$

C-Oct.93 Karpovsky et al Fig. 11. 3-Dim P-ary Array of PEs

Array	Parameters (N, n, d)	Estimates of a Minimal Number of Signatures r Lower Bound	Upper Bound	Compression Rate $\frac{r}{n}(n \to \infty)$
Line of length n (Fig. 5a)	(n, n, n)	n	n	1
2-dim $(h \times w)$mesh (Fig. 5b)	$(hw, h+w-1, h+w-1)$	$h+w-1$	$h+w-1$	1
p-ary star $p \geq 2$	$(p+1, p, 2)$	$\lceil \log_2(p+2) \rceil$	$\lceil \log_2(p+2) \rceil$	$\frac{\log_2 p}{p}$
Balanced binary tree with d levels	$(2^d-1, 2^{d-1}, d)$	min r: $\sum_{i=1}^{r-d+1} \binom{r}{i} \geq 2^{d-1}$ (Table 2)	$6\lceil \frac{d-1}{4} \rceil$ (Table 2)	$1.5\frac{\log_2 n}{n}$
Balanced p-ary tree with d levels $(p \geq 2)$	$(\frac{p^d-1}{p-1}, p^{d-1}, d)$	min r: $\sum_{i=1}^{r-d+1} \binom{r}{i} \geq p^{d-1}$ and $\lceil \log_2 \left(\frac{p^d-1}{p-1}+1\right) \rceil$ (Table 3)	$(d-1)\lceil \log_2(p+2) \rceil$ (Table 3)	$2\frac{\log_2 n}{n}$
$(d \times n)$rhombic mesh (Fig. 6a)	(dn, n, d)	min r: $\sum_{i=1}^{\lceil \frac{r-d}{2} \rceil} \binom{r}{i} + \sum_{i=\lceil \frac{r+d}{2} \rceil}^{r} \binom{r}{i} \geq n$ and $\lceil \log_2(nd+1) \rceil$ (Tables 4 and 5)	$\lceil \log_2 \frac{n}{3(d+1)} \rceil + d + 3$ (Tables 4 and 5)	$\frac{\log_2 n}{n} + \frac{d}{n}$
Triangular mesh (Fig. 10)	$(0.5(d+1)d, 2d-1, d)$	$d+1$	$d+1$	0.5
3-dim p-ary cube (Fig. 11)	$(p^3, 3p^2-3p+1, 3p-2)$	$3p-2$	$3p-2$	$\frac{1}{p}$
m-dim binary cube	$(2^m, 2^m-1, m+1)$	$m+1$	$m+1$	$\frac{m+1}{2^m-1}$
m-dim p-ary cube	$(p^m, p^m-(p-1)^m, (p-1)m+1)$	$(p-1)m+1$	$(p-1)m+1$	$\frac{1}{p^{m-1}}$

C-Oct. 93 Karpovsky et al. Table VI

C-Oct 93
Karpovsky et al.
Table VIII

d	η_2	η_3
2	.667	.000
3	.571	.257
4	.438	.1473
5	.368	.0843
6	.202	.0279
7	.176	.0121
8	.114	.0076

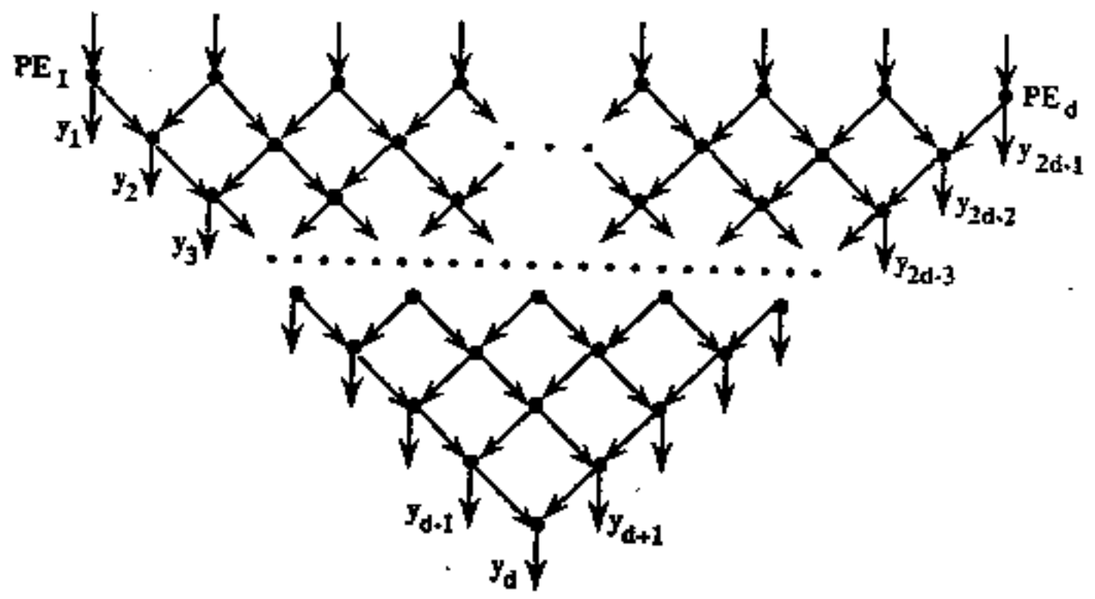
75%

75%

d,n	η_2	η_3
2,3	0.400	0.000
2,9	0.301	0.039
2,18	0.248	0.031
2,36	0.201	0.042
3,12	0.292	0.041
3,24	0.230	0.026
4,15	0.283	0.039
4,30	0.219	0.023
5,18	0.257	0.037
5,36	0.211	0.021
6,21	0.268	0.035
7,24	0.264	0.033
8,27	0.259	0.032
9,30	0.256	0.031
10,33	0.253	0.029

C-Oct. 93
Karpovsky et al
Table VIII

50% ($\frac{k}{2d-1}$)

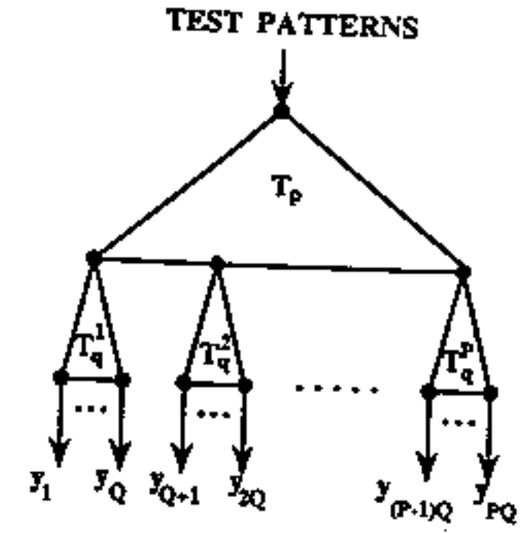


n=2d-1 ; N= $\frac{1}{2}$d(d+1)

Error vectors for d=4:
(1111000), (0111100), (0011110), (0001111)
(0111000), (0011100), (0001110)
(0011000), (0001100)
(0001000)

C-Oct .93 Karpovsky et al Fig.10. Triangular Mesh of PEs

50%

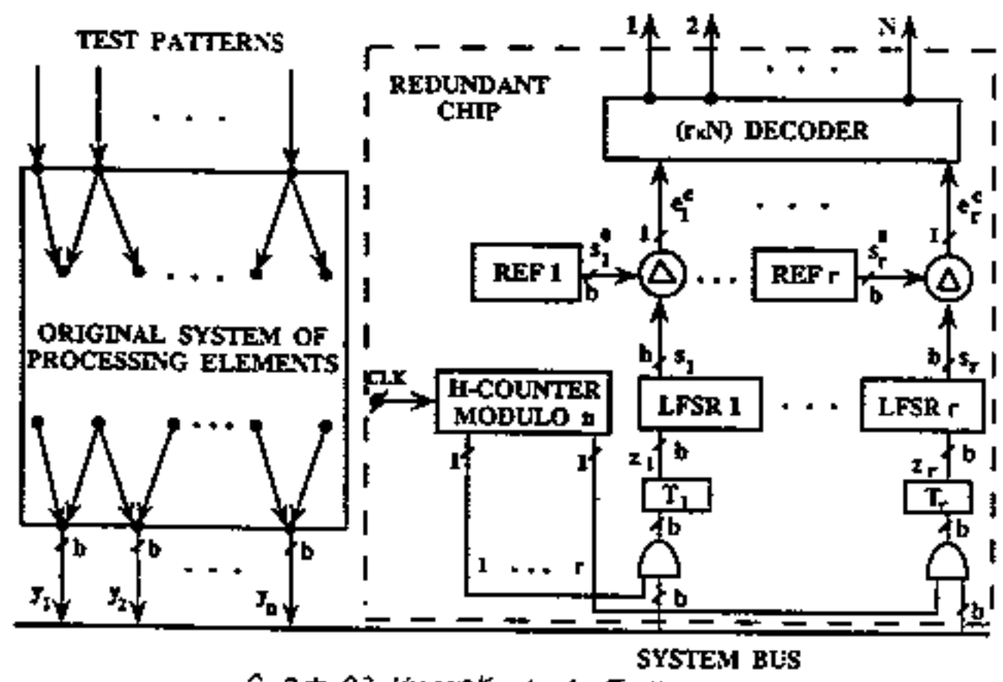


d=p+q-1; P=2^{p-1}; Q=2^{q-1}; n=2^{d-1}=PQ

$T_q^i=T_q$ (i=1,...,P); N=2^d-1

C-Oct.93 Karpovsky et al Fig 7

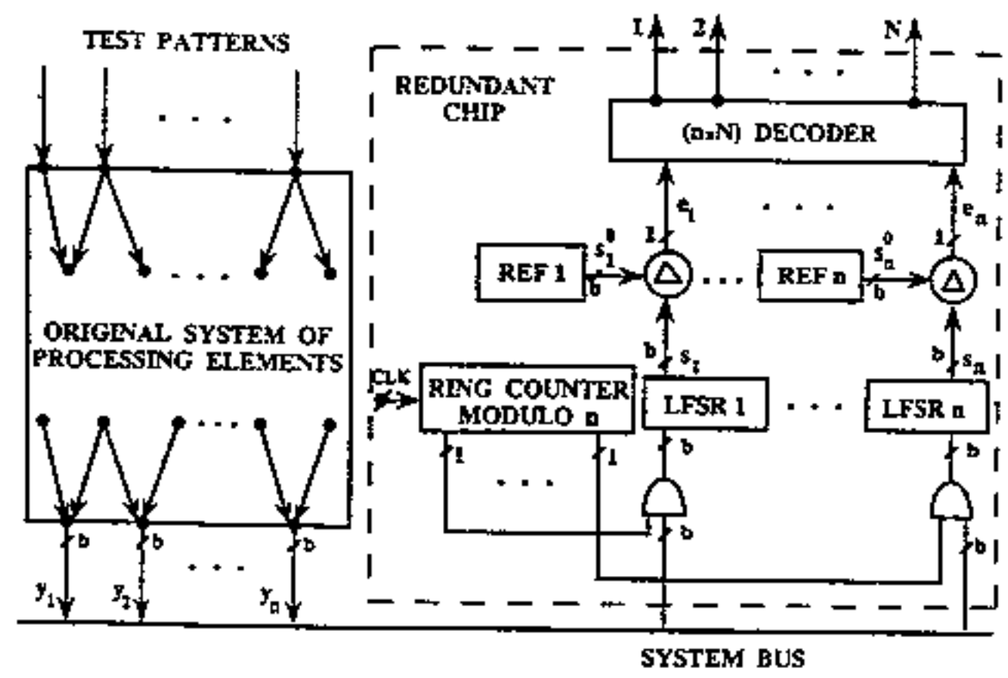
42%



C-Oct. 93 Karpovsky et al. Fig 4

Fig.4. Space-Time Approach for Diagnostics

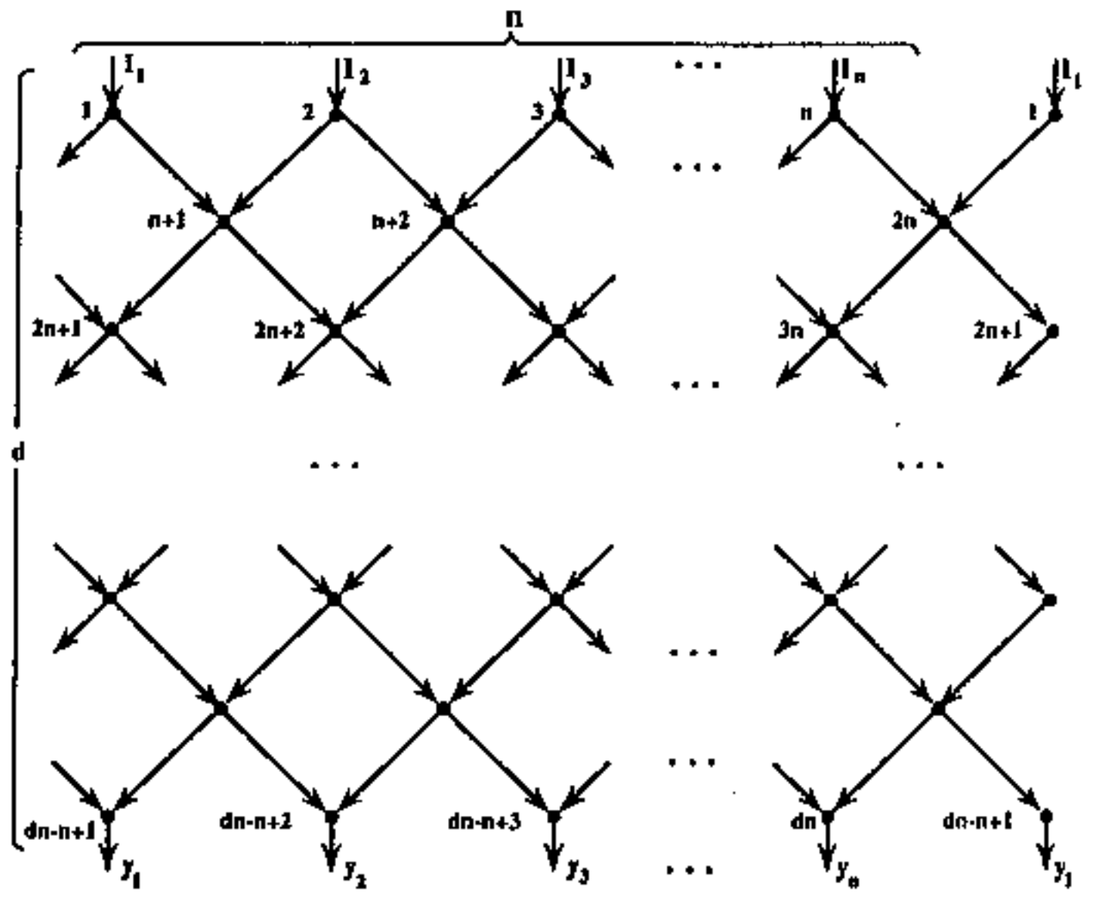
42%



C-Oct. 93 Karpovsky et al.

Fig.1. The Straightforward Approach to Diagnostics

45% (fit x21)

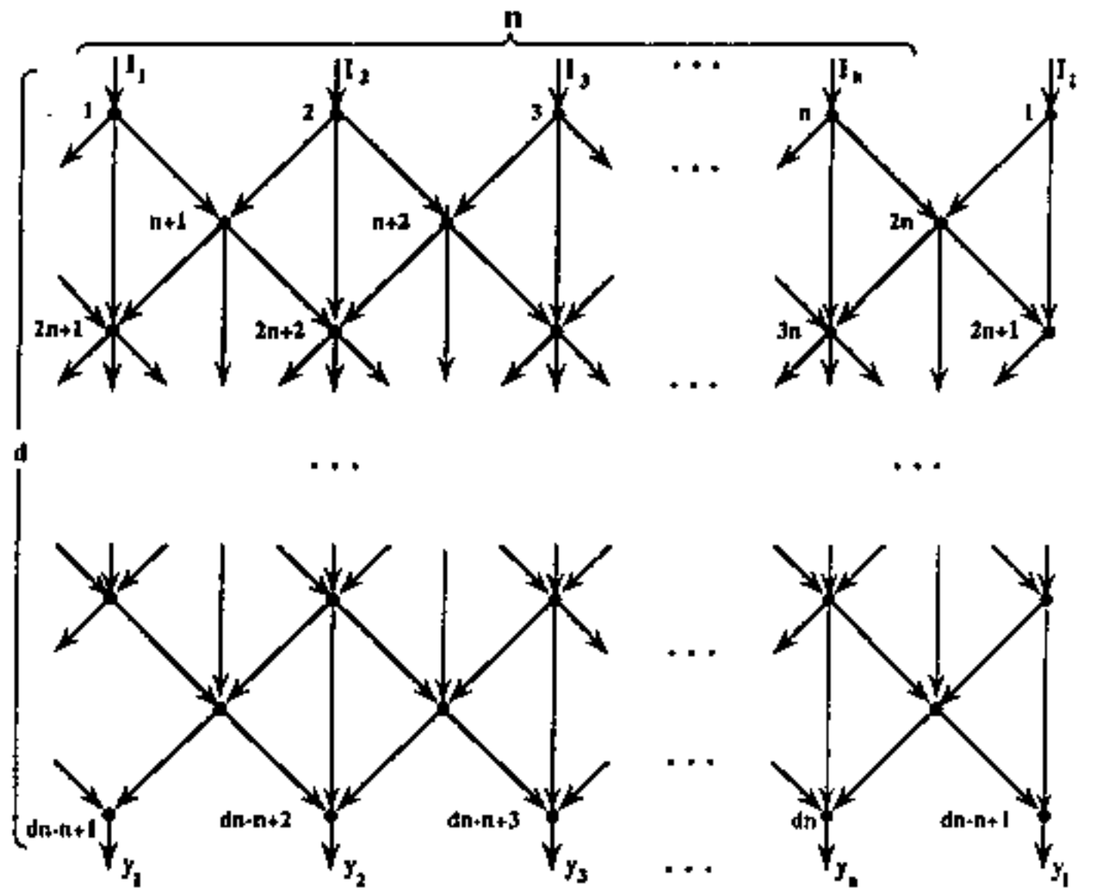


Error vectors for d=3 and n=5:
(11001), (11100), (01110), (00111), (10011)
(10001), (11000), (01100), (00110), (00011)
(10000), (01000), (00100), (00010), (00001)

C-Oct.93 Karpovsky et al. Fig 6(a)

Fig.6a. (dxn) Rhombic Mesh of PEs

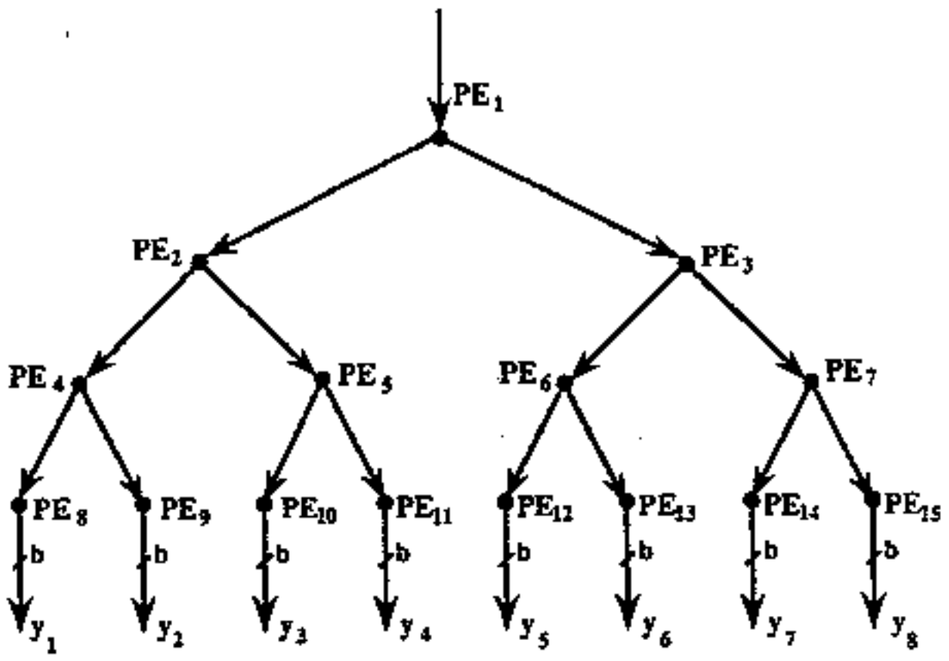
45% (fit x21)



Error vectors for d=3 and n=5:
(11001), (11100), (01110), (00111), (10011)
(10001), (11000), (01100), (00110), (00011)
(10000), (01000), (00100), (00010), (00001)

C-Oct.93 Karpovsky et al. Fig. 6(b)

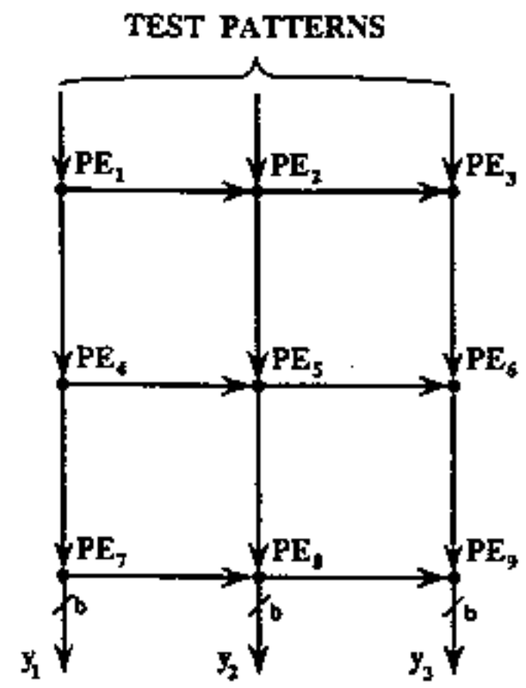
Fig.6b. Hexagonal Mesh of PEs



C-Oct-93 Karpovsky et al.

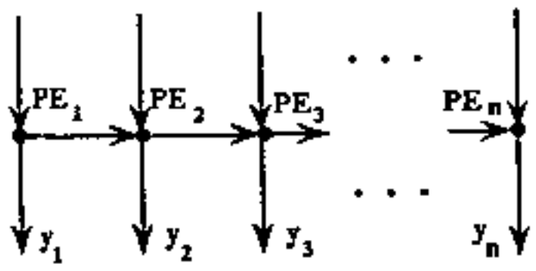
Fig.2. Three-Level Balanced Tree of PEs

50%



C-Oct. 93 Karpovsky et al. Fig.3

50%



r=d=n=N

Error vectors for n=4:
(1111), (0111), (0011), (0001)

(a)

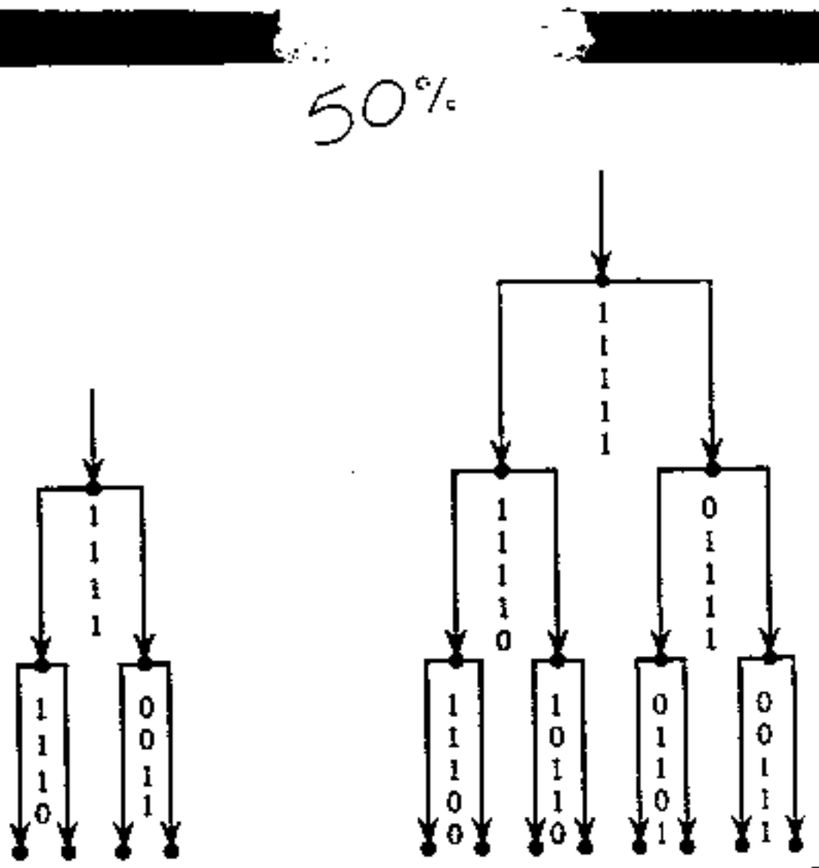
r=d=n=h+w-1

N=hw

Error vectors for h=w=3:
(11111) (01111) (00111)
(11110) (01110) (00110)

Fig. 5

50%

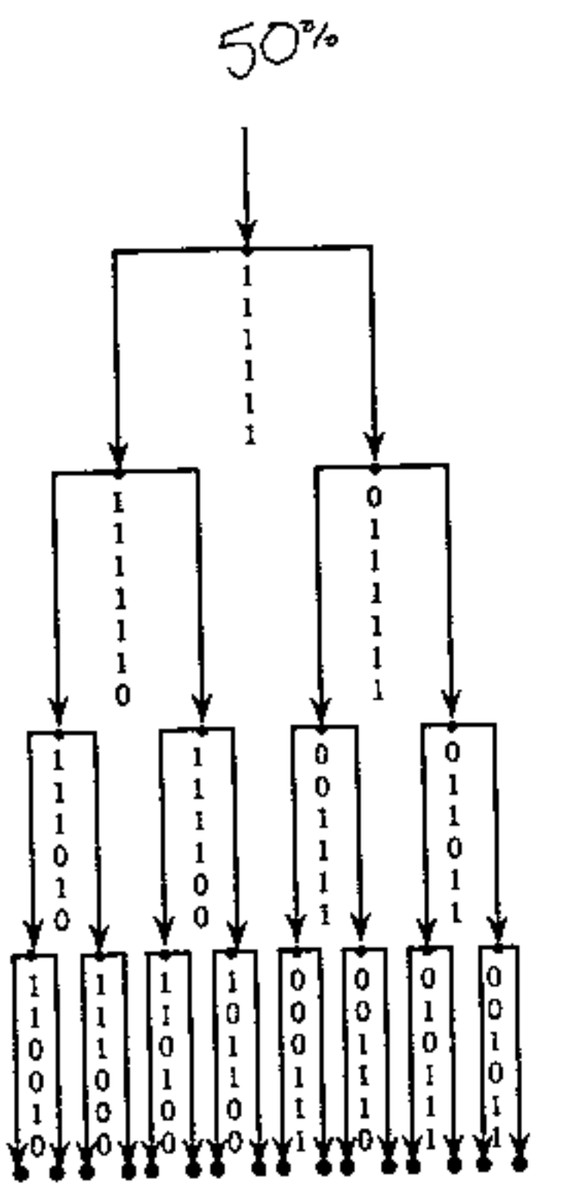


$$H_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad H_4 = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

$$H_6 = \begin{bmatrix} 1111\ 0000\ 1111\ 1111\ 0000\ 0000\ 0000\ 0000 \\ 1111\ 1111\ 0000\ 0000\ 1111\ 0000\ 0000\ 0000 \\ 0000\ 1111\ 1111\ 0000\ 0000\ 1111\ 1111\ 0000 \\ 0000\ 0000\ 0000\ 1111\ 0000\ 0000\ 1111\ 1111 \\ 0000\ 0000\ 0000\ 0000\ 1111\ 0000\ 0000\ 1111 \\ \\ 1000\ 1000\ 1000\ 1000\ 1000\ 1000\ 1000\ 1000 \\ 1100\ 1100\ 1100\ 1100\ 1100\ 1100\ 1100\ 1100 \\ 0101\ 0101\ 0101\ 0101\ 0101\ 0101\ 0101\ 0101 \\ 0010\ 0010\ 0010\ 0010\ 0010\ 0010\ 0010\ 0010 \end{bmatrix}$$

C-Oct 93 Karpovsky et al Fig 9

50%



$$H_5 = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

C-Oct. 93 Karpovsky et al. Fig 8

Fig.8. Optimal Space Compression Matrix H_5 and Embedding of the Five-Level Binary Tree into 6-Dimensional Binary Cube