Identification of Faulty Processing Elements by Space–Time Compression of Test Responses [1]

M. G. Karpovsky L. B. Levitin F. S. Vainstein

Research Laboratory of Design and Testing of Computer Hardware
Department of Electrical, Computer and Systems Engineering
Boston University
College of Engineering
44 Cummington Street
Boston, Ma. 02215

Abstract

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

1 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., systolic array). The system is represented by a directed graph G whose nodes correspond to processing elements (PEs) and directed edges corresponds to communication links. We assume that at most one PE in the system may be faulty. 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 straightforward approach to diagnostics by signature analysis is illustrated with Fig. 1. Test responses $y(t) = (y_1(t), \ldots, y_n(t))$ at moment t ($y_i(t)$ is a b-bit binary vector) are transferred via system bus into a redundant chip in such a way that the test response $y_i(t)$ at the output i is compressed in time by Linear Feedback Shift Register (LFSR) i. After all test responses $y(1), \ldots, y(T)$ (T is the number of test responses) have been compressed by the LFSRs, the corresponding signatures $s_1, \ldots, s_n$ are compared with the precomputed reference signatures $s_1^0, \ldots, s_n^0$, and the error vector $e = (e_1, \ldots, e_n)$ is computed, where

$$e_i = \begin{cases} 1 & s_i \neq s_i^0 \\ 0 & s_i = s_i^0 \end{cases} \qquad (1)$$

The identification of a faulty PE is implemented by the $n \times N$ decoder (N is the total number of PEs in the system) with the input $e = (e_1, \ldots, e_n)$. We assume that a number of test responses T is sufficiently large, so that a fault in a PE will manifest itself by distortions of signatures corresponding to all output PEs connected with the faulty PE.

For example, if the original array is a binary balanced tree (Fig. 2), a fault in PE_2 will result in error vector (11110000) (we assume that the fault is not masked in any one of the 8-bit LFSRs compressing in time $y_i(1), \ldots, y_i(T)$ ($i = 1, \ldots, 8$). The probability of masking is very small for large b). The relation between faulty PEs and error vectors for the binary tree of Fig. 2 is given in the first two columns of Table 1.

The system is diagnosable iff all the n-bit error vectors are different and not equal to $(0, \ldots, 0)$. An example of a nondiagnosable system is given at Fig. 3. In this case faults in PE_2 and PE_5 cannot be distinguished, since in both cases $e = (011)$. Thus, we have the following lower bound on a number of outputs n of a diagnosable system

$$n \geq \lceil \log_2(N+1) \rceil, \qquad (2)$$

where N is the total number of PEs in the system.

Note that the lower bound in (2) is attainable. To demonstrate this, let us consider an array which is the n-dimensional binary cube with one node being deleted ($N = 2^n - 1$). In this case PEs are numbered by nonzero n-bit binary vectors and there is a directed edge from $u = (u_1, \ldots, u_n)$ to $v = (v_1, \ldots, v_n)$ iff the Hamming distance between u and v is equal to one and $u_i \geq v_i$ ($i = 1, \ldots, n; u_i, v_i \in \{0, 1\}$). Let us assume that outputs of the system are taken from the PEs numbered by n-bit vectors of weight 1 (i.e., having one nonzero component) (Fig. 4 shows the system for n = 3). Then, $n = \lceil \log_2(N+1) \rceil$, and it is clear that the number of a faulty PE can be computed as $(100\ldots0) \cdot e_1 \vee (01\ldots0) \cdot e_2 \vee$

[1] This work has been partially supported by NSF under Grant MIP 8813748

$\ldots \vee (00\ldots1)\cdot e_n$, where $\vee$ stands for the component-wise OR operation.

It is worth to note also that the reason for considering single faults only is that by analyzing error vectors e we cannot distinguish between single and some double faults. For example, for the binary tree of Fig. 2 one cannot distinguish between a double fault in PE_2 and PE_5 and a single fault in PE_2.

For the straightforward approach to diagnostics represented at Fig. 1, the required hardware overhead L_1, in terms of a number of equivalent two-input gates, is of the order of $L_1 = O(bn)$. For example, for the eight-level binary tree with b = 32 we have n = 128, N = 255 and $L_1 \simeq 110,000$ (assuming that one flipflop is equivalent to 8 gates). In this paper another approach to diagnostics will be suggested which results in a considerable reduction of the required overhead while the probability of missing a fault remains small. We will see below that in many cases the overhead can be decreased to $L_2 = O(b\log_2 n)$.

To illustrate this approach let us return back to the example of three-level binary tree with $n = 8$, $N = 15$ (Fig. 2). Instead of compressing in time the sequence $y(1),\ldots,y(T)$ (where $y(t) = (y_1(t),\ldots,y_8(t))$ and $y_i(t)$ is b-bit binary vector) by 8 LFSRs, we first compute $z(t) = Hy(t)$, where

$$H = \begin{pmatrix} 01000100 \\ 10100000 \\ 00010001 \\ 00001010 \\ 00100111 \end{pmatrix}, \qquad (3)$$

and all the computations are made modulo two (this is the space compression step). Then $z(t) = (z_1(t),\ldots,z_5(t))$. Now, we will compress in time the sequence $z(1),\ldots,z(T)$ using only 5 LFSRs. The resulting 5 signatures $s_1,\ldots,s_5$ are compared with the precomputed 5 reference values $s_1^0,\ldots,s_5^0$, and the identification of a faulty PE is made by analyzing the error syndrome (the compressed error vector) $e^c = (e_1^c,\ldots,e_5^c)$, 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 (see Fig. 2), then one can see that $e^c = (01101)$. Error syndromes e^c for different faults are presented in the rightmost column of Table 1. Since different faults results in different nonzero syndromes e^c, identification of a faulty PE can be implemented by decoding e^c. Thus we have been able to reduce an overhead (using only 5 LFSR and 5 reference values, instead of 8 for the original approach) and still we can identify a faulty PE.

The block-diagram for the proposed diagnostic approach with space-time compression is given at Fig. 5. The output response vector $y(t) = (y_1(t),\ldots,y_n(t))$ is compressed in space into $z(t) = (z_1(t),\ldots,z_r(t))$ where $y_i(t)$ and $z_j(t)$ are binary vectors, and $z(t) = Hy(t)$ and H is a binary $(r\times n)$-matrix $(r \leq n)$. This space compression is implemented by an H-counter modulo n. The sequence of output vectors for this counter is the sequence of r-bit columns of matrix H.

Table 1: Relation Between Faulty PE's, Error Vectors and Error Syndromes For the Three-Level Binary Tree

Faulty PE	Error Vector e	Error Syndrome e^c
1	11111111	11111
2	11110000	11101
3	00001111	10111
4	11000000	11000
5	00110000	01101
6	00001100	10011
7	00000011	00111
8	10000000	01000
9	01000000	10000
10	00100000	01001
11	00010000	00100
12	00001000	00010
13	00000100	10001
14	00000010	00011
15	00000001	00101

Space signatures $z(t) = (z_1(t),\ldots,z_r(t))$ are compressed in time by r LFSRs. Final space-time signatures $s_1,\ldots,s_r$ are compared with the precomputed reference values $s_1^0,\ldots,s_r^0$, and the resulting error syndrome $e^c = (e_1^c,\ldots,e_r^c)$ ($e_i^c = 1$ iff $s_i = s_i^0$) is decoded to indicate the faulty processor. This identification is possible iff there is a one-to-one mapping between PEs and error vectors $e^c = (e_1^c,\ldots,e_r^c)(e_i^c \in \{0,1\})$. This mapping means an embedding of the graph G representing original system of PEs 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 y to be a descendant of vector x, if y can be obtained from x by replacing some of the components equal to 1 by zeros. (It is said also that x covers y.) The embedding of graph G into the r-dimensional cube must preserve the partial ordering on G defined by the directed edges. The embedding of the three-level binary tree into 5-dimensional binary cube is given by the rightmost column of Table 1.

An overhead for the space-time compression is of the order of $L_2 = O(br)$ and comparing with the overhead L_1 for the straightforward approach we have

$$\frac{L_1}{L_2} \simeq \frac{n}{r}. \qquad (4)$$

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

Since all error syndromes must be different and not equal to $(0,\ldots,0)$ we have the following attainable bounds

$$\lceil \log_2(N+1) \rceil \leq r \leq n. \qquad (5)$$

The overhead minimization problem for the space-time signature diagnostics can be reduced to constructing an $(r\times n)$ matrix H with minimal r such that the system remains di-

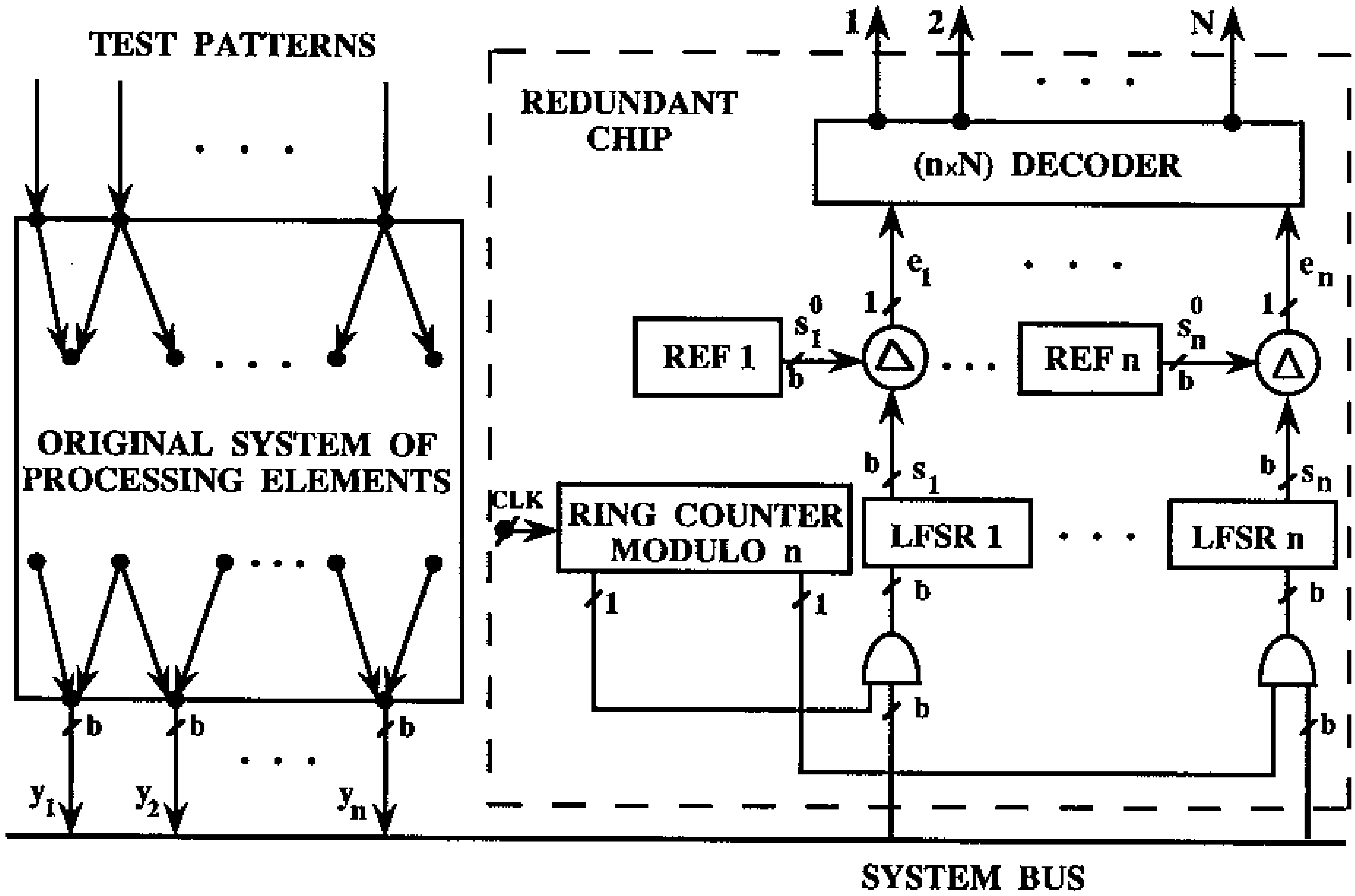


Fig.1. The Straightforward Approach to Diagnostics

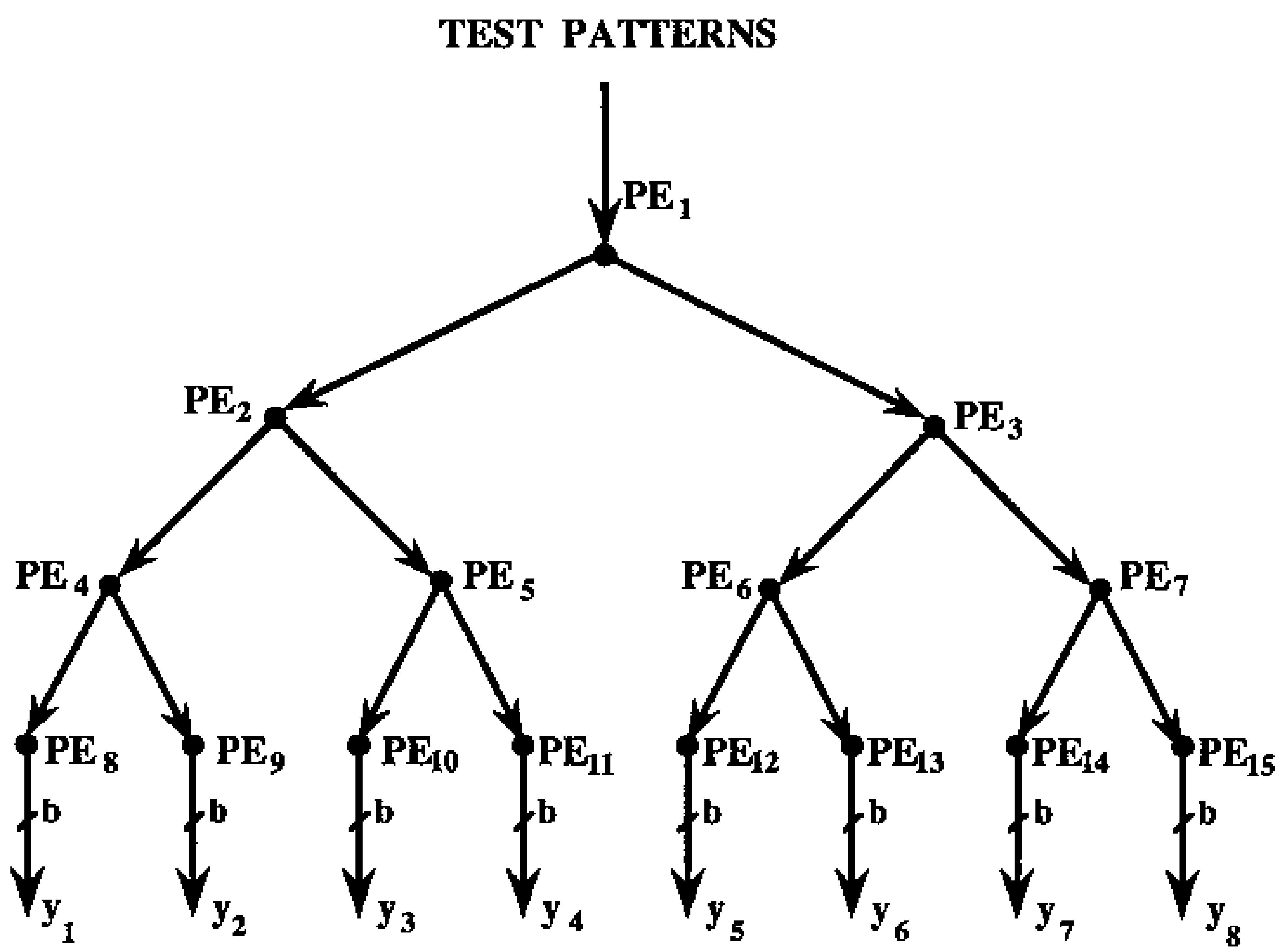


Fig.2. Three-Level Balanced Tree of PEs

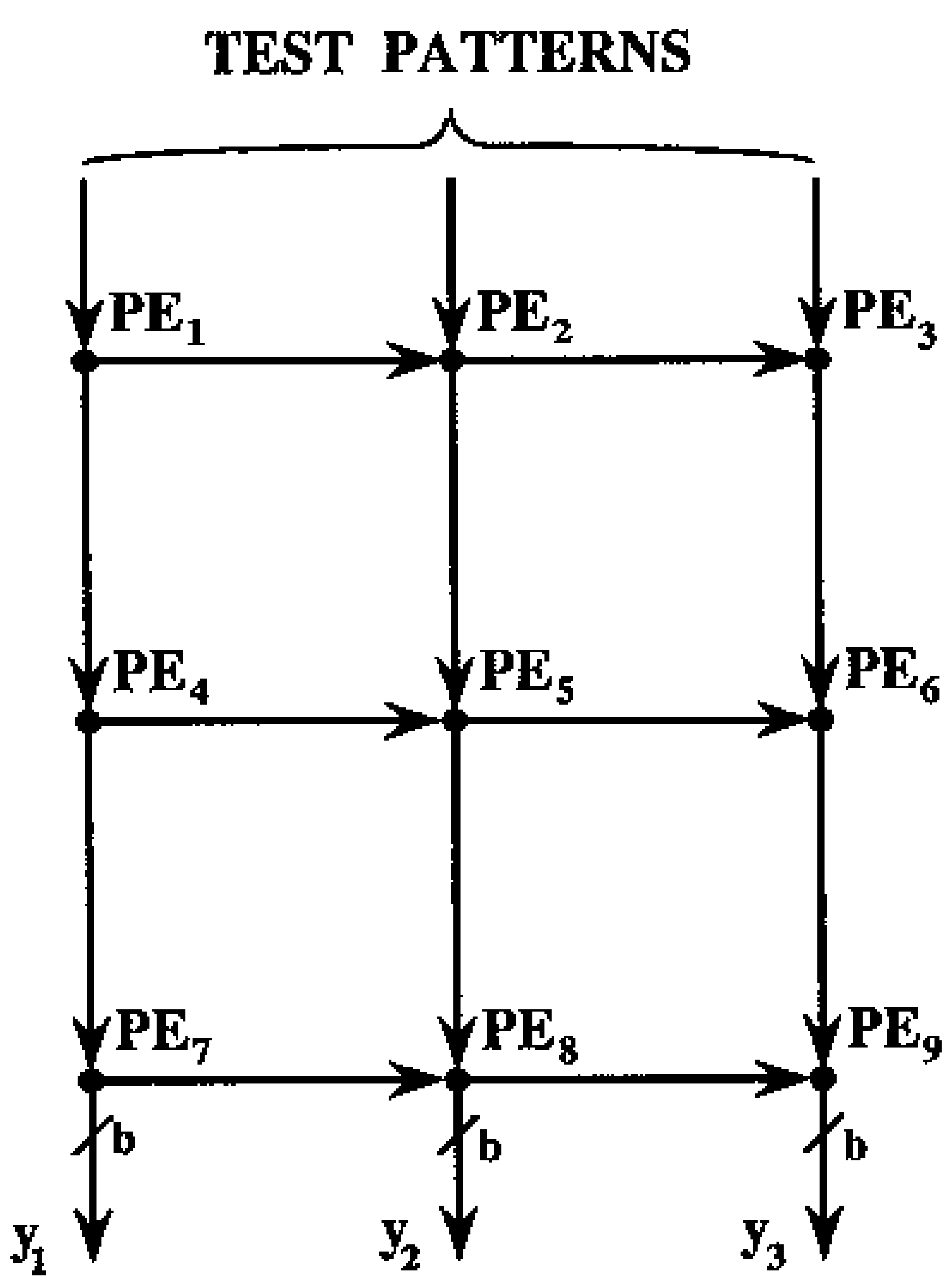


Fig. 3. An example of Nondiagnozable Systolic Array

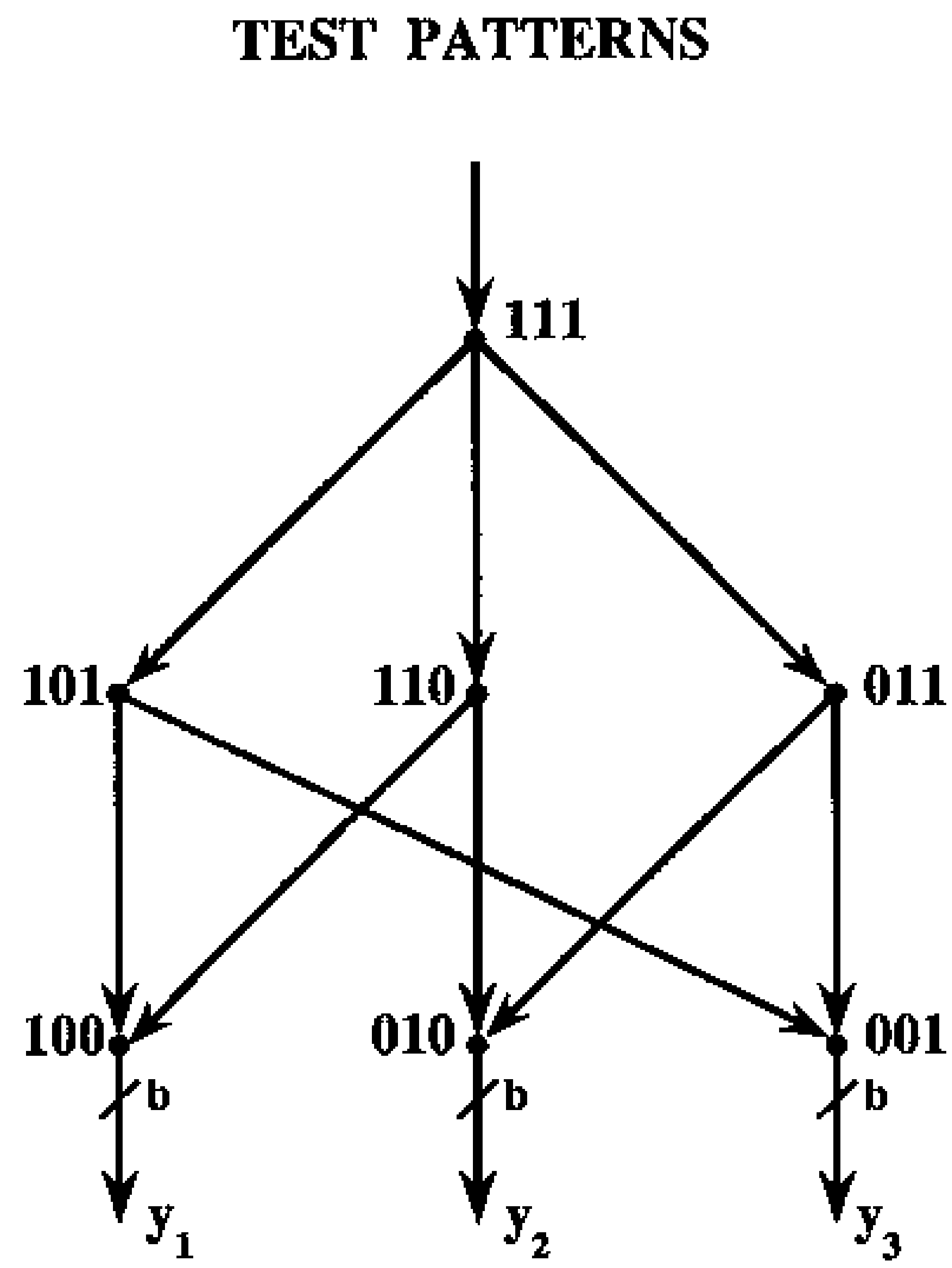


Fig. 4. 3-Cube of PEs with one PE detected

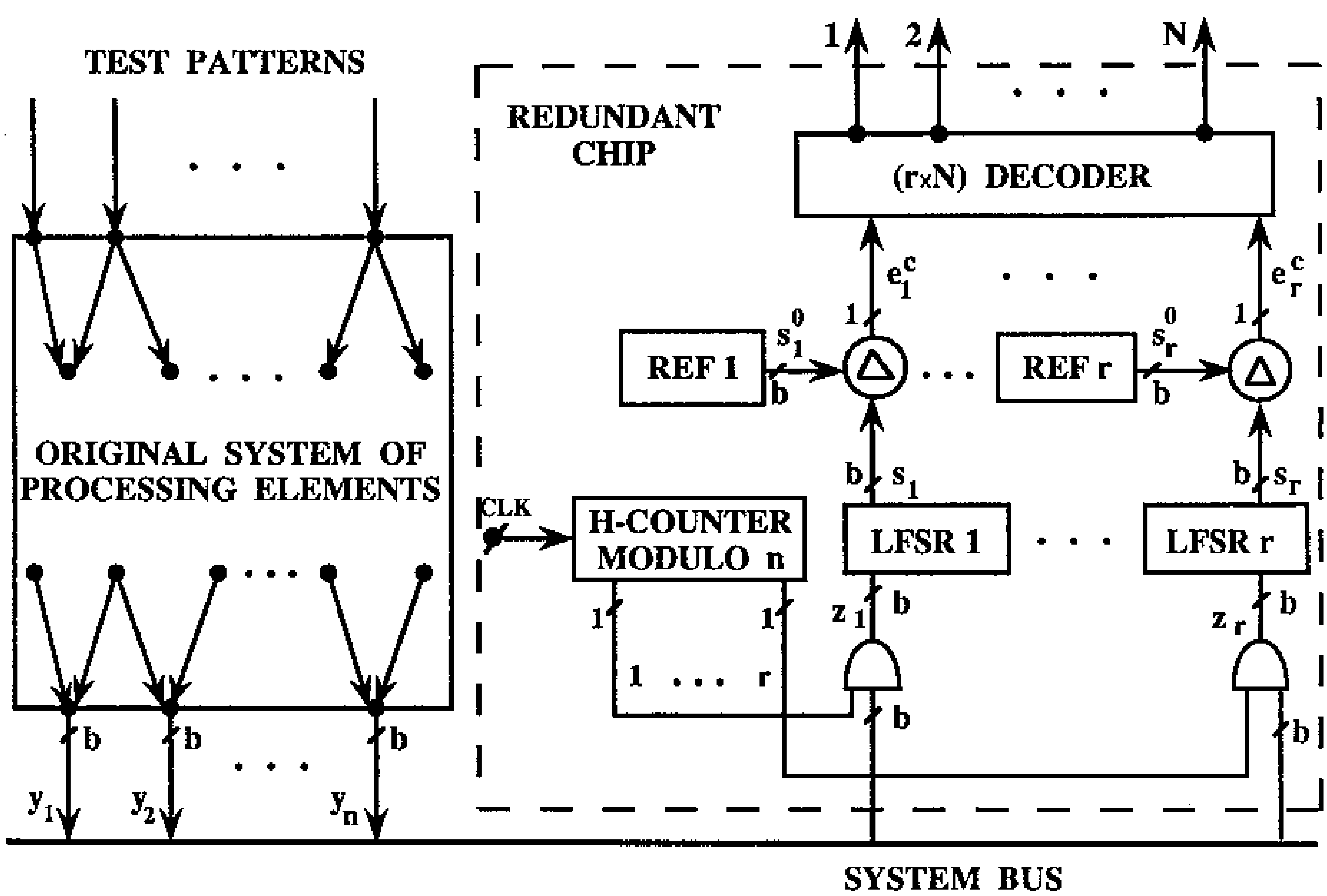


Fig.5. Space-time Approach for Diagnostics

agnosable after the space compression $z(t) = Hy(t)$ of its output $y(t)$.

It is easy to show, that the relation between the error vectors e in the original system and the error syndrome e^c is given by the following formula:

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

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

$$e_c = \begin{pmatrix} 01000100 \\ 10100000 \\ 00010001 \\ 00001010 \\ 00100111 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \tag{7}$$

which corresponds to the fifth row in Table 1.

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'

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

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.

The solution for the overhead minimization problem for several important classes of systems is given in the next section.

The proposed space-time signature approach to diagnostics is based on the "hard decision" decoding of signatures $s = (s_1, \ldots, s_r)$, when we can identify a faulty PE by analyzing binary vector e^c which indicates the distorted component in s. The magnitudes of distortions are not important for the hard decision procedure. One can use a "soft decision decoding" of $s = (s_1, \ldots, s_r)$ 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 [11,12] for board-level space-time signature diagnosis and in [15] for space-time diagnosis of multiprocessor systems. In [11-12] and [13] the assumption have 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 that PEs are disconnected in the testing mode.

2 Hardware Minimization for Space-Time Signature Diagnosis

It was 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 low bound for r. Suppose the maximum number of PEs in a path from input PE to an output PE (depth of the system) is d. Then for embedding the system of PEs into r-cube,

$$r \geq d. \tag{9}$$

This is an attainable lower bound, which can be illustrated by examples of a line array (Fig. 6a) and of the two-dimensional near-neighbour mesh (Fig. 6b).

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 two important classes of systems: balanced binary tree with $n = 2^{d-1}$, $N = 2^d - 1$ (see Fig. 2 for $n = 8$) and rhombic meshes (see Fig. 7). These arrays as well as n-cubes, lines and 2-d near-neighbour meshes have been widely used [14].

2.1 Space–Time Diagnosis for Balanced Binary Trees

For the d-level binary tree T_d (d is the number of PEs on the the path from the input to any output, $n = 2^{d-1}$, $N = 2^d - 1$) we denote by $r(d)$ the minimal number of signatures to be stored, i.e. $r(d)$ is minimal dimension of 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 1 we have $r(4) \leq 5$.

Let us derive a lower bound for $r(d)$ which is better than the general bounds (5) and (9). Since in T_d there are $n = 2^{d-1}$ paths from the input to n outputs, for embedding T_d into $C_{r(d)}$ the output PEs should be encoded by different nonzero r(d)-dimensional binary vectors of weight at most $r(d) - d + 1$. Thus

$$\sum_{i=1}^{r(d)-d+1} \binom{r(d)}{i} \geq 2^{d-1}. \tag{10}$$

Solving (10) for large d we have

$$r(d) > 1.29(d-1). \tag{11}$$

To construct $r \times n$ 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 balanced binary tree T_d represented in Fig. 8. Here $d = p + q - 1$ and $T_q^1, T_q^2, \ldots, T_q^P$ $(P = 2^{p-1})$ are identical trees T_q of depth q and $Q = 2^{q-1}$ 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 easy to show that H_d can be constructed as

$$H_d = \begin{pmatrix} h_p^1 h_p^1 \cdots h_p^1 & h_p^2 h_p^2 \cdots h_p^2 & \cdots & h_p^P h_p^P \cdots h_p^P \\ h_q^1 h_q^2 \cdots h_q^Q & h_q^1 h_q^2 \cdots h_q^Q & \cdots & h_q^1 h_q^2 \cdots h_q^Q \end{pmatrix}. \quad (12)$$

Thus, we have

$$r(d) = r(p + q - 1) \leq r(p) + r(q). \quad (13)$$

An example of this construction for $d = 6$, $p = 4$ and $q = 3$ is shown at 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 1. From (3) an (10) we have $r(4) = 5$, which shows that the lower bound given by (10) is attainable.

Matrix H_5 and the corresponding embedding of T_5 into C_5 is given in Fig. 10. By Fig.10 and (10), $r(5) = 6$. Using this result and (13) we obtain:

$$r(d) \leq 6 \left\lceil \frac{d-1}{4} \right\rceil, \quad (14)$$

which is close to lower bounds (10) and (11). Some exact values of $r(d)$ and upper and lower bounds are given in Table. 2.

Table 2: Minimal Numbers of Signatures $r(d)$ Required for Diagnostics of d-Level Binary Trees

d	2	3	4	5	6	7	8
$r(d)$	2	4	5	6	8	9-10	10-11

d	9	10	11	12
$r(d)$	12	13-14	14-16	16-17

Table 2 illustrates 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 2), we have a reduction in hardware (measured in equivalent two-input gates) from $L_1 = 110,000$ to $L_2 = 10,000$.

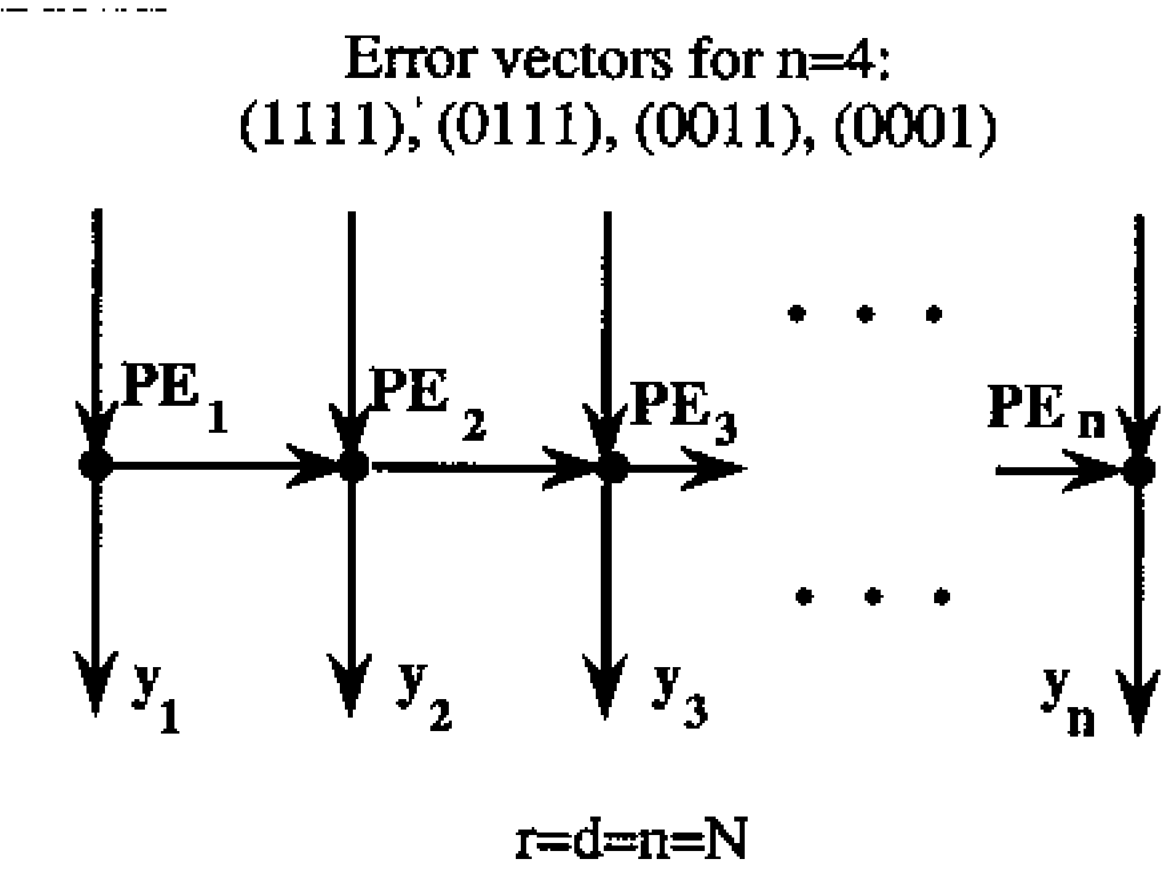


Fig.6a. Line of PEs

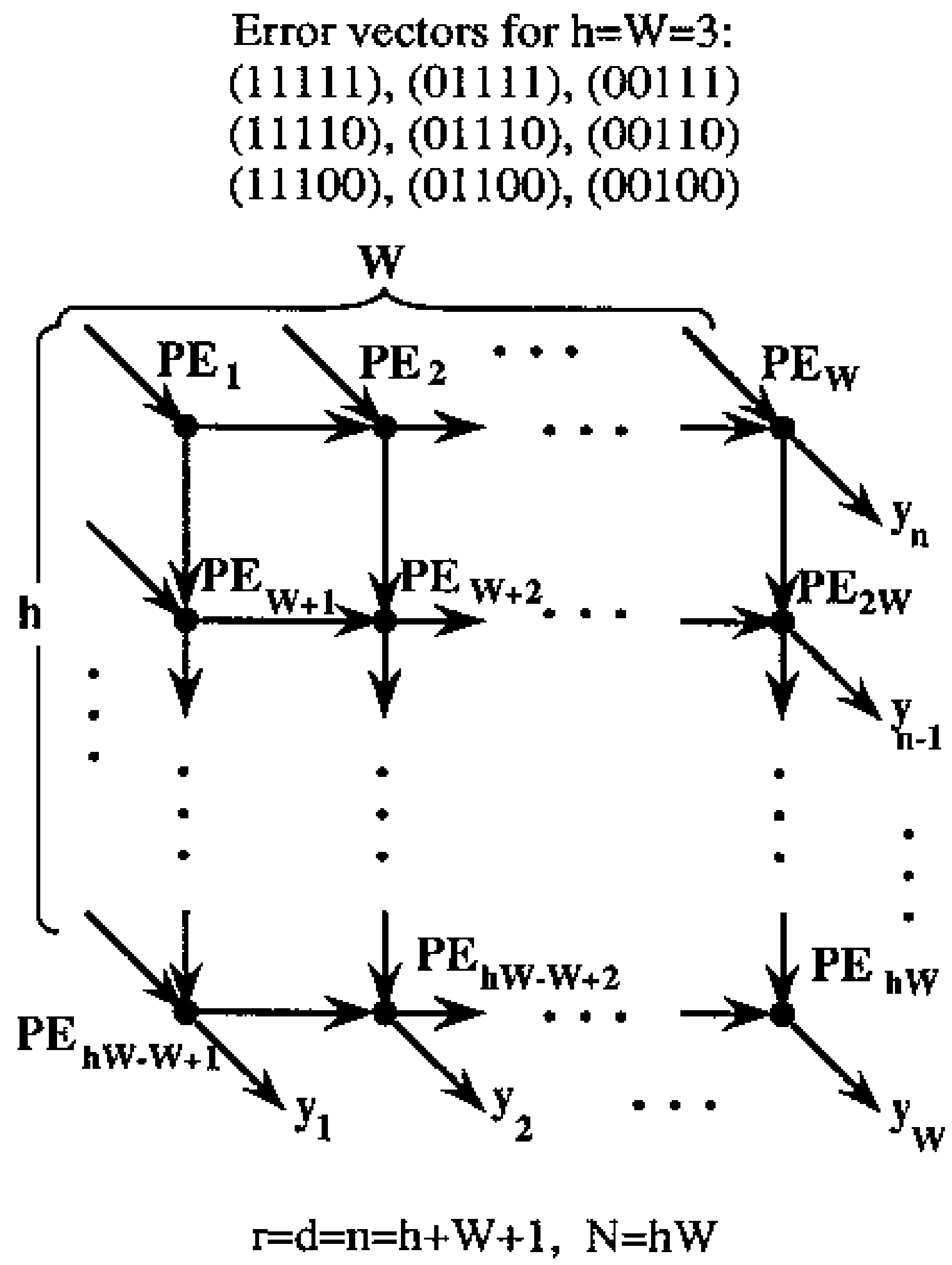


Fig.6b. (h×W) Near-Neighbour Mesh of PEs

To conclude this section we note that lower bounds (10), (11), upperbounds (13), (14) and construction (12) for H_d can be generalized for non-binary trees.

2.2 Space–Time Signature Diagnosis for Rhombic Mesh Arrays

The cylindrical rhombic mesh is shown in Fig. 7. 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 (9) are

$$r(n,d) \geq d \text{ and} \quad (15)$$

$$r(n,d) \geq \lceil \log_2(nd+1) \rceil. \quad (16)$$

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 nodes in common.

Let our array be embedded into an r-dimensional binary cube. Then each path includes d nodes and the binary r-dimensional vector corresponding to a node is a descendant of the vector corresponding to the previous node in the path. Thus, each path contains vectors of d different weights, and the difference between the weights of the endpoints of each path is at least $d - 1$.

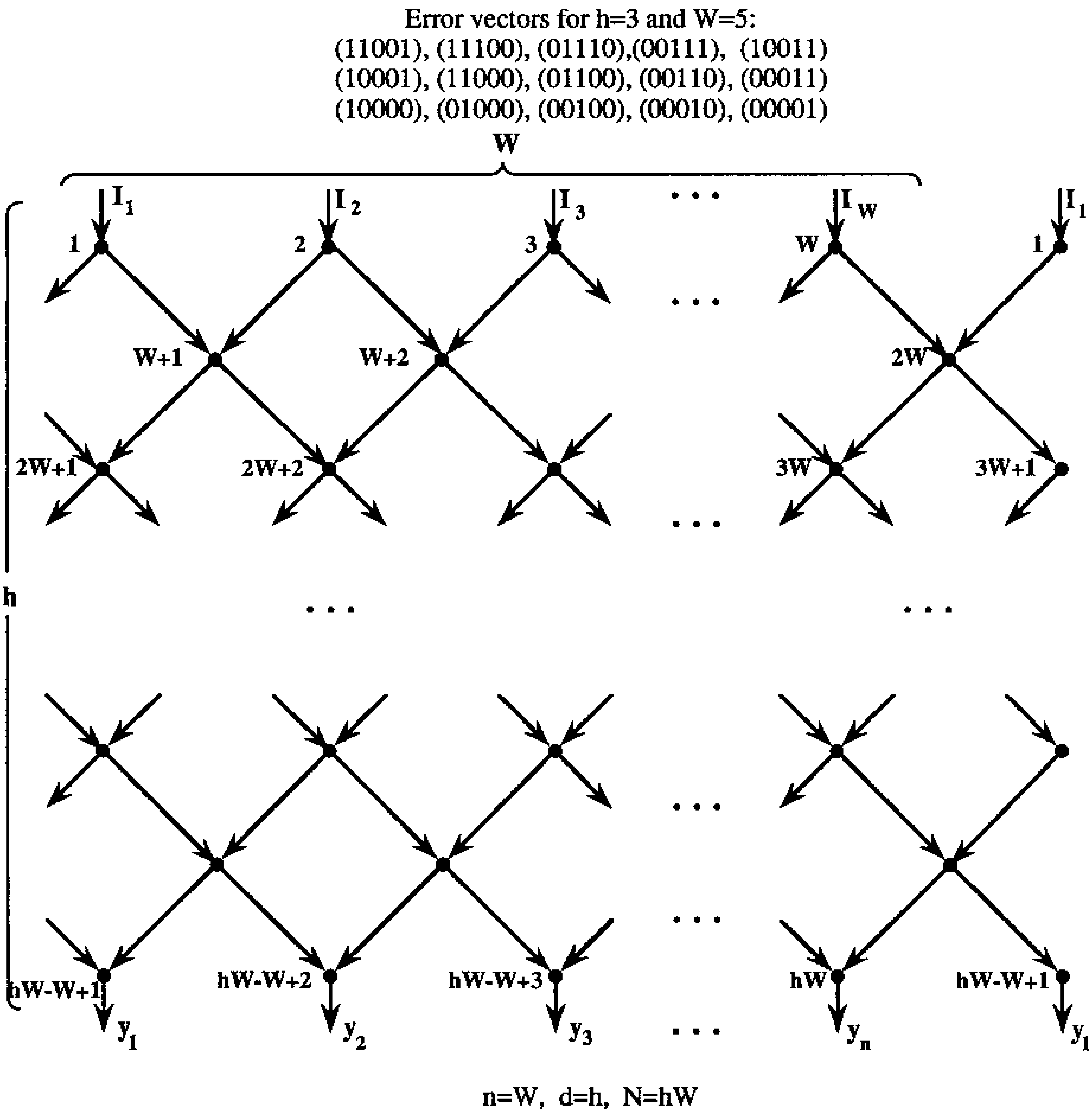


Fig.7. Cylindrical (h×W) Rhombic Mesh of PEs

Consider now two "polar zones" in the cube: the vectors of weight $w \geq \lceil \frac{r+d}{2} \rceil$ and of weight $w \leq \lceil \frac{r-d}{2} \rceil$. It is easy to see that at least one of the endpoints of each of the above-mentioned paths belongs to one of the polar zones. Therefore

$$n \leq \sum_{i=1}^{\lceil \frac{r-d}{2} \rceil} \binom{r}{i} + \sum_{i=\lceil \frac{r+d}{2} \rceil}^{r} \binom{r}{i}. \qquad (17)$$

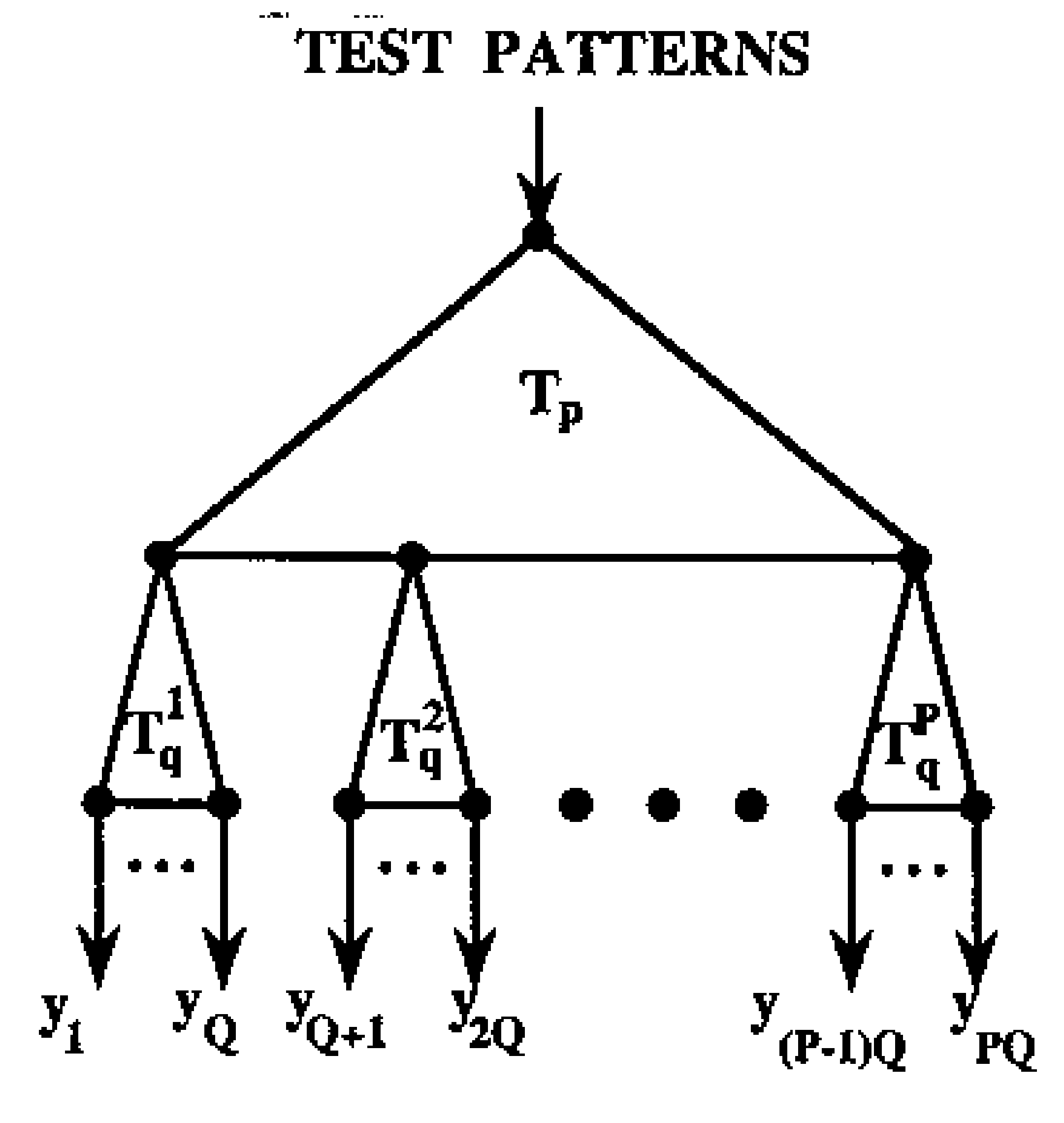
The minimum value of r that satisfies (17) is the lower bound for $r(n,d)$. The lower bound given by (17) always superseeds the bound (15), but (16) still provides a better lower bound for very large values of n ($\log_2 n \gg d^2$).

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:

$$H_1 = \begin{pmatrix} I_{d+1} & I_{d+1} & I_{d+1} \\ 00\cdots0 & 00\cdots0 & 11\cdots1 \\ 00\cdots0 & 11\cdots1 & 00\cdots0 \end{pmatrix}$$
$$H_2 = \begin{pmatrix} I_{d+1} & I_{d+1} & I_{d+1} \\ 11\cdots1 & 00\cdots0 & 00\cdots0 \\ 00\cdots0 & 11\cdots1 & 00\cdots0 \end{pmatrix}, \qquad (18)$$

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

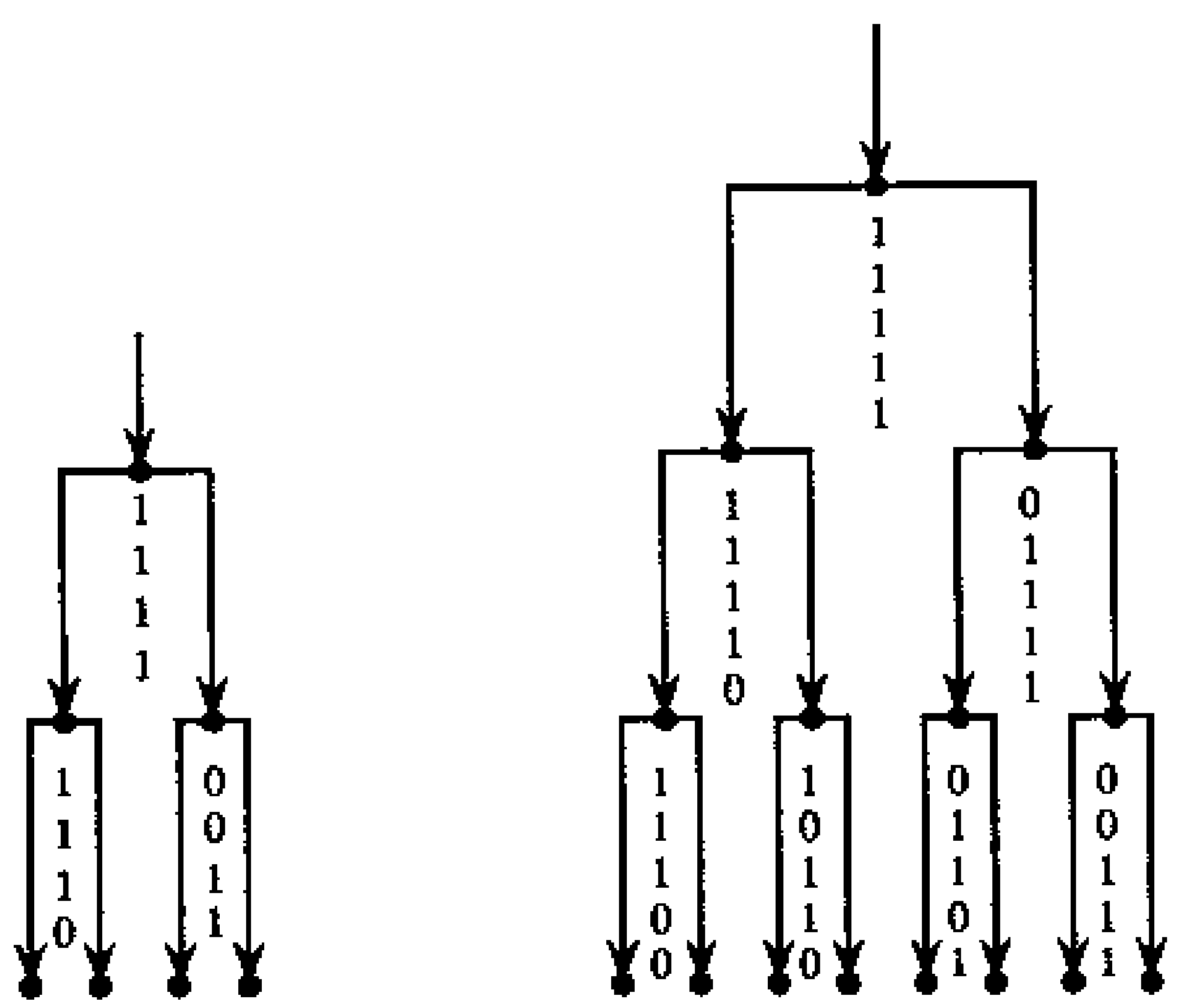
Now let $n = 3(d+1)m$ and $k = \lceil \log_2 m \rceil$, where $m = 1, 2, \ldots$. Let g_l be the codeword for the integer l ($0 \leq l \leq 2^k - 1$) in the k-bit Gray (reflexive) code. Denote by A_l a $k \times 3(d+1)$ matrix which consists of identical columns g_l. Let B_l be the $(k+d+3) \times 3(d+1)$ matrix which is obtained by vertical concatenation (writing one matrix under the other) of matrices H_1 and A_l for an even l and H_2 and A_l for odd



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

$T_q^i=T_q$ $(i=1,\dots,P)$

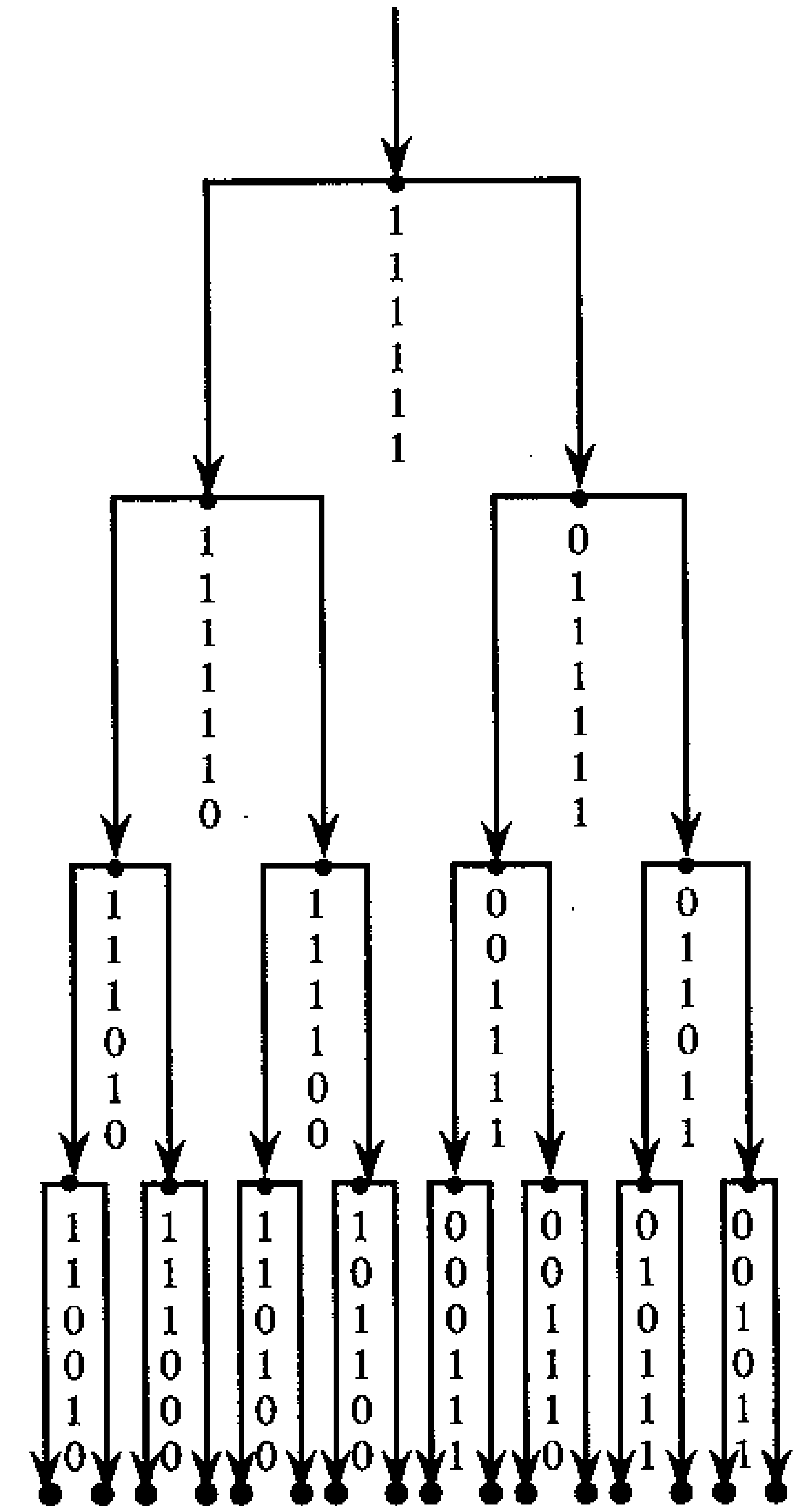
Fig.8. Recursive Construction for d-level Binary Tree T_d



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

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

Fig.9. Construction of the Space Compression Matrix for the Binary Trees T_3, T_4, and T_6 (d=3,4,6)



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

Fig.10. Optimal Space Compression Matrix H_5 and Embedding of the Five-Level Binary Tree into 6-Dimensional Binary Cube

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 $(l = 0, 1, \dots, m-1)$. Thus, the number of rows in H is given by

$$r = k + d + 3 = \lceil \log_2 \frac{n}{3(d+1)} \rceil + d + 3, \quad \textit{and} \qquad (19)$$

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

An example of matrix H for $n = 18$, $d = 2$ is given below:

$$H = \begin{pmatrix} 100 & 100 & 100 & 100 & 100 & 100 \\ 010 & 010 & 010 & 010 & 010 & 010 \\ 001 & 001 & 001 & 001 & 001 & 001 \\ 000 & 000 & 111 & 111 & 000 & 000 \\ 000 & 111 & 000 & 000 & 111 & 000 \\ 000 & 000 & 000 & 111 & 111 & 111 \end{pmatrix}$$

It can be readily shown, that all the syndromes obtained by H designed above are different. Indeed, two errors within the same block b_l are distinguished by matrices H_1 or H_2. Two errors within different blocks B_{l_1} and B_{l_2} are distinguished by the matrices A_{l_1} and A_{l_2}, since OR of two consecutive codewords of the Gray code gives always one of these words.

If two errors belong to two disjoint pair of blocks B_{l_1}, B_{l_1+1} and B_{l_2}, B_{l_2+1}, respectively, their syndromes will differ in some of the last k digits. The last possible case is when two errors belong to overlapping pairs of blocks B_l, B_{l+1} and B_{l+1}, B_{l+2}, respectively. Then their syndromes will differ in the $(d+2)$th digit.

Matrices H for $n \neq 3(d+1)m$ can be obtained by slight modifications of the construction given above. We note also that formula (19) remain valid for $n = d+1$, and $H = I_{d+1}$.

The lower bounds given by (15) and (16) are attainable and sometimes coinside 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 $n = 3(d+1)m$:

$$r(n,2) = \lceil \log_2 m \rceil + 5 = \lceil \log_2 n - 2\log_2 3 \rceil + 5, \quad (20)$$

$$r(n,3) = \lceil \log_2 m \rceil + 6 = \lceil \log_2 n - \log_2 3 \rceil + 4, \quad (21)$$

$$\lceil \log_2 m \rceil + 6 \leq r(n,4) \leq \lceil \log_2 m \rceil + 7, \quad (22)$$

$$\lceil \log_2 m \rceil + 7 \leq r(n,5) \leq \lceil \log_2 m \rceil + 8, \quad (23)$$

$$d+3 \leq r(6(d+1),d) \leq d+4, \quad (24)$$

$$r(3(d+1),d) = d+3, \quad (25)$$

$$r(d+1,d) = d+1, \quad (26)$$

$$r(d+2,d) = d+2. \quad (27)$$

The lower bound (L) based on (15), (16) and (17) and the upper bound (U) based on (19) for $r(n,d)$ are presented in Table 3. for some $n \leq 2000$ and $d \leq 20$. Results for small n and d are shown in Table 4.

Expression (19) shows that space-time signature diagnostics provides considerable hardware savings as compared to the straightforward approach (time compression only). For example, for a 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.

Table 3: Bounds on the Minimal Numbers $r(n,d)$ of Signatures for Rhombic $(n \times d)$-Meshes.

d = 2			d = 3			d = 4			d = 5		
n	L	U	n	L	U	n	L	U	n	L	U
9	5	5	12	6	6	15	7	7	18	8	8
18	6	6	24	7	7	30	7	8	36	8	9
36	7	7	48	8	8	60	8	9	72	9	10
72	8	8	96	9	9	120	9	10	144	10	11
144	9	9	192	10	10	240	10	11	288	11	12
288	10	10	384	11	11	480	11	12	576	12	13

d = 6			d = 7			d = 8			d = 9		
n	L	U	n	L	U	n	L	U	n	L	U
21	9	9	24	10	10	27	11	11	30	12	12
42	9	10	48	10	11	54	11	12	60	12	13
84	10	11	96	11	12	108	12	13	120	13	14
168	11	12	192	12	13	216	13	14	240	14	15
336	12	13	384	13	14	432	13	15	480	14	16
672	13	14	768	14	15	864	14	16	960	15	17

d = 10			d = 11			d = 12			d = 13		
n	L	U	n	L	U	n	L	U	n	L	U
33	13	13	36	14	14	39	15	15	42	16	16
66	13	14	72	14	15	78	15	16	84	16	17
132	14	15	144	15	16	156	16	17	168	17	18
264	15	16	288	16	17	312	17	18	336	18	19
528	15	17	576	16	18	624	17	19	672	18	20
1056	16	18	1152	17	19	1248	18	20	1344	19	21

d = 14			d = 15			d = 16			d = 17		
n	L	U	n	L	U	n	L	U	n	L	U
45	17	17	48	18	18	51	19	19	54	20	20
90	17	18	96	18	19	102	19	20	108	20	21
180	18	19	192	19	20	204	19	21	216	20	22
360	19	20	384	20	21	408	20	22	432	21	23
720	19	21	768	20	22	816	21	23	864	22	24
1440	20	22	1536	20	23	1632	21	24	1728	22	25

d = 18			d = 19			d = 20		
n	L	U	n	L	U	n	L	U
57	21	21	60	22	22	63	23	23
114	21	22	120	22	23	126	23	24
228	21	23	240	22	24	252	23	25
456	22	24	480	23	25	504	24	26
912	23	25	960	24	26	1008	25	27
1824	23	26	1920	24	27	2016	25	28

Table 4: Minimal Numbers of Signatures r(n, d) for Rhombic (n x d)-Meshes with Small n and d.

d	2	3	4	5	6	7	8	9	10
n	L-U	L-U	L-U	L-U	L-U	L-U	L-U	L-U	L-U
3	3	-	-	-	-	-	-	-	-
6	4	5	6	6	-	-	-	-	-
9	5	5-6	6-7	7-8	8-9	9	9	-	-
12	5	6	6-7	7-8	8-9	9-10	10-11	11-12	12
15	5-6	6-7	7	7-8	8-9	9-10	10-11	11-12	12-13
18	6	6-7	7-8	8	8-9	9-10	10-11	11-12	12-13
21	6	6-7	7-8	8-9	9	10	10-11	11-12	12-13
24	6-7	7	7-8	8-9	9-10	10	11	12	12-13
27	6-7	7-8	7-8	8-9	9-10	10-11	11	12	13
30	6-7	7-8	8	8-9	9-10	10-11	11-12	12	13
33	7	7-8	8-9	8-9	9-10	10-11	11-12	12-13	13
36	7	7-8	8-9	8-9	9-10	10-11	11-12	12-13	13-14

3 Conclusion

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 LFSRs 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.

References

[1] Smith J. E., "Measures of the Effectiveness of Fault Signature Analysis," IEEE Trans. on Computers, 1980, c-29, pp510-516

[2] McAnney W. H. and Savir J., "There is Information in Faulty Signatures," Proc 1987 Int. Test Conference, Sept 1987, pp 630-636

[3] Savir J. and McAnny W. H., "Identification of Failing Test with Cyclic Registers," Proc 1988 Int. Test Conference, Sept 1988, pp 322-328

Konemann B, Mucha J and G. Zwieholf, "Built-in Test for Complex Digital Integrated Circuits," IEEE J. Solid State Circuits, Vol SC-15 No 3, June 1980, pp 315-318

Upadhyaya S. J. and Saluja K. K., "Signature Techniques in Fault Detection" in "Spectral Techniques and Fault Detection" M. Karpovsky editor, Academic Press, 1985, pp 421-477

Bardell P. H. and McAnney W. H., "Self-Testing of Multichip Logic Modules," Proc. Int. Test Conference 1982, pp 200-204

Williams T. W., Daehn W., Gruetzner M. and Starke C. W., "Bounds and Analysis of Aliasing Errors in Linear Feedback Shift Registers," IEEE Trans. Computer-Aided Design, Vol 7, Jan 1988, pp 75-83

Ivanov A. and Agarwal V. K., "An Analysis of the Probabilistic Behavior of Linear Feedback Shift Registers," IEEE Trans. Computer-Aided Design, Vol 8, No 10, Oct 1989, pp 1074-1088

Pradhan D. K., Gupta S. K. and Karpovsky M. G., "Aliasing Probability for Multiple Input Signature Analyzer," IEEE Trans on Computers, April 1980

Reddy S. R., Suluja K. K. and Karpovsky M. G., "A Data Compression for Built-In Self Test," IEEE Trans. on Computers, Sept 1988, pp 1151-1156

Karpovsky M. G. and Nagvajara P., "Board Level Diagnosis," Proc. Int. Test Conference, 1988, pp 47-53

[12] Karpovsky M. G. and Nagvajara P., "Design of Self-Diagnostic Boards by Signature Analysis," IEEE Trans. on Industrial Engineering, April 1989

[13] Karpovsky M. G., "An Approach for Error Detection and Error Correction in Distributed Systems Computing Numerical Functions," IEEE Trans. on Computers, Dec 1981, No 12, pp 947-954

[14] Hwang K., Briggs F. A., "Computer Architecture and Parallel Processing," McGraw Hill, 1984