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

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

Research Laboratory of Design and Testing of Computer Hardware
Department of Electrical, Computer and Systems Engineering
Boston University
College of Engineering
44 Cammington 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 diagnosis.

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 whose nodes correspond to processing elements (PEs) and directed edges correspond 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 $y(t) = (y_1(t), ..., y_n(t))$ at moment $t$ ($y(t)$ is a b-bit binary vector) are transferred via system bus into a redundant chip in such a way that the test response $y(t)$ at the output $i$ is compressed in time by Linear Feedback Shift Register (LFSR) $i$. After all test responses $y(1), ..., y(T)$ ($T$ is the number of test responses) have been compressed by the LFSRs, the corresponding signatures $s_1, ..., s_n$ are compared with the precomputed reference signatures $s'_1, ..., s'_n$, and the error vector $e = (e_1, ..., e_n)$ is computed, where

$$
e_i = \begin{cases} 1 & s_i \neq s'_i \\ 0 & s_i = s'_i \end{cases}$$

The identification of a faulty PE is implemented by the $\times N$ decoder ($N$ is the total number of PEs in the system) with the input $e = (e_1, ..., e_n)$. 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 $y_1(1), ..., y_8(T)$ ($t = 1, ..., 8$). 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 $(0, ..., 0)$. An example of a nondiagnosable system is given at Fig. 3. In this case faults in $PE_3$ 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

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

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 $u = (u_1, ..., u_n)$ to $v = (v_1, ..., v_n)$ if the Hamming distance between $u$ and $v$ is equal to one and $u_i \neq v_i$ ($i = 1, ..., n; u_i, v_i \in \{0, 1\}$).

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 $100 \cdot \cdot \cdot e_1 \cdot \cdot \cdot 010 \cdot \cdot \cdot e_3 \cdot$
... V (00...) \cdot \varepsilon, \text{ where } V \text{ 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 \( \varepsilon \) 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 \( PE_b \) and \( PE_a \) and a single fault in \( PE_a \).

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 \( L_1 \approx 110,000 \) (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 \( y(t), y(T) \) (where \( y(t) = (y_1(t), \ldots, y_6(t)) \) and \( y_T \) is \( b \)-bit binary vector) by \( 8 \) LFRSs, we first compute \( x(T) = HY(t), \) where

\[
H = \begin{pmatrix}
00100100 \\
10101000 \\
00010100 \\
00001010 \\
00000101 \\
00000011
\end{pmatrix},
\]

and all the computations are made modulo two (this is the space compression step). Then \( x(t) = (x_1(t), \ldots, x_6(t)) \).

Now, we will compress in time the sequence \( x(1), \ldots, x(T) \) using only \( 8 \) LFRSs. The resulting signatures \( s_1, \ldots, s_6 \) are compared with the precomputed reference values \( s_1^*, \ldots, s_6^* \), and the identification of a faulty PE is made by analyzing the error syndrome (the compressed error vector) \( s^* = (s_1^*, \ldots, s_6^*) \), where \( s_i^* = 1 \) iff \( s_i \neq s_i^* \) and \( s_i^* = 0 \), otherwise. For example, if \( PE_a \) is faulty (see Fig. 2), then one can see that \( s^* = (001101) \).

Error syndromes \( s^* \) for different faults are presented in the rightmost column of Table 1. Since different faults result in different nonzero syndromes \( s^* \), identification of a faulty PE can be implemented by decoding \( s^* \). Thus we have been able to reduce an overhead (using only 8 LFRS and 6 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 \( y(t) = (y_1(t), \ldots, y_6(t)) \) is compressed in space into \( x(t) = (x_1(t), \ldots, x_6(t)) \) where \( y(t) \) and \( x(t) \) are binary vectors, and \( x(t) = HY(t) \) 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>
<thead>
<tr>
<th>Faulty PE</th>
<th>Error Vector ( \varepsilon )</th>
<th>Error Syndrome ( s^* )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11111111</td>
<td>11111</td>
</tr>
<tr>
<td>2</td>
<td>11110000</td>
<td>11101</td>
</tr>
<tr>
<td>3</td>
<td>00001111</td>
<td>10111</td>
</tr>
<tr>
<td>4</td>
<td>11000000</td>
<td>11000</td>
</tr>
<tr>
<td>5</td>
<td>00110000</td>
<td>01101</td>
</tr>
<tr>
<td>6</td>
<td>00001100</td>
<td>10011</td>
</tr>
<tr>
<td>7</td>
<td>00000011</td>
<td>00111</td>
</tr>
<tr>
<td>8</td>
<td>10000000</td>
<td>00000</td>
</tr>
<tr>
<td>9</td>
<td>01000000</td>
<td>10000</td>
</tr>
<tr>
<td>10</td>
<td>00100000</td>
<td>001001</td>
</tr>
<tr>
<td>11</td>
<td>00010000</td>
<td>001000</td>
</tr>
<tr>
<td>12</td>
<td>00001000</td>
<td>000100</td>
</tr>
<tr>
<td>13</td>
<td>00000100</td>
<td>10001</td>
</tr>
<tr>
<td>14</td>
<td>00000010</td>
<td>000101</td>
</tr>
<tr>
<td>15</td>
<td>00000001</td>
<td>001011</td>
</tr>
</tbody>
</table>

Space signatures \( x(t) = (x_1(t), \ldots, x_6(t)) \) are compressed in time by \( r \) LFRSs. Final space-time signatures \( s_1, \ldots, s_6 \) are compared with the precomputed reference values \( s_1^*, \ldots, s_6^* \), and the resulting error syndrome \( s^* = (s_1^*, \ldots, s_6^*) \) iff \( s_i = s_i^* \) is decided to indicate the faulty processor. This identification is possible iff there is a one-to-one mapping between PEs and error vectors \( s^* = (s_1^*, \ldots, s_6^*) \) (where \( i \in \{0, 1\} \)).

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 \( z \), if \( y \) can be obtained from \( z \) by replacing some of the components equal to 1 by zeros. (It is said also that \( z \) 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 \( L_2 = O(bn) \) and comparing with the overhead \( L_1 \) for the straightforward approach we have

\[
L_2 \simeq \frac{n}{r}
\]

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 \( s^* \).

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

\[
[\log_b(N + 1)] \leq r \leq n
\]

The overhead minimisation problem for the space-time signature diagnostics can be reduced to constructing an \((nxn)\)-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 Non-diagnosable Systolic Array

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

Fig. 5. Space-time Approach for Diagnostics
agnosable after the space compression \( s(t) = H y(t) \) 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 \( e' \) is given by the following formula:

\[
    e' = H \otimes e
\]

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 \( P E_6 \) being faulty we have from (3) \( e = (00110000) \) and

\[
    e_6 = \begin{pmatrix}
    0 & 0 & 1 & 0 & 1 & 0 \\
    0 & 0 & 0 & 1 & 0 & 1 \\
    0 & 0 & 0 & 0 & 1 & 0
    \end{pmatrix}
\]

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 \( e' \):

\[
    H \otimes e \neq H \otimes e', H \otimes e \neq 0, H \otimes e' \neq 0.
\]

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 \( s = (s_1, \ldots, s_n) \), when we can identify a faulty PE by analyzing binary vector \( e' \) 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 \( s = (s_1, \ldots, s_n) \) 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 a cube,

\[
    r \geq d.
\]

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. 3 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 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 \( C_{2^d} \) such that \( T_d \) can be embedded in \( C_{2^d} \) 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 (9) and (9). Since in \( T_d \) there are \( n = 2^d - 1 \) paths from the input to \( n \) outputs, for embedding \( T_d \) into \( C_{2^d} \) the output PEs should be encoded by different nonzero \( r(d) \)-dimensional binary vectors of weight at most \( r(d) - d + 1 \). Thus

\[
    \sum_{i=1}^{r(d) - d + 1} r(d)_i \geq 2^d - 1.
\]

Solving (10) for large \( d \) we have

\[
    r(d) > 1.29(d - 1).
\]

To construct \( r \times n \) space compression matrix \( H_{r, n} \) for \( T_d \) (which yields an embedding of the \( T_d \) into \( C_n \) and provides an upper bound for \( r(d) \)) we will use the recursive construction for balanced binary tree \( T_d \) represented in Fig. 9. Here \( d = p + q - 1 \) and \( T_{d_1} \), \( T_{d_2}^1 \), \( T_{d_2}^2 \) \( (P = 2^q - 1) \) 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 \( H_p = [h_1^p, h_2^p, \ldots, h^p] \) and \( H_q = [h_1^q, h_2^q, \ldots, h^q] \), respectively, where \( h_1^p \) and \( h_1^q \) 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_1^p & h_2^p & \cdots & h^p \\
  h_1^q & h_2^q & \cdots & h^q \\
   &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &   &   \\
  &   &  \ldots
\end{pmatrix}
\tag{12}

Thus, we have

\[
r(d) = r(p + q - 1) \leq r(p) + r(q). \tag{13}
\]

An example of this construction for \( d = 6, p = 4 \) and \( q = 3 \) is shown at Fig. 9.

Matrix \( H_d \) is given by (3) and the corresponding embedding of \( T_p \) into \( C_6 \) 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 \( H_d \) and the corresponding embedding of \( T_p \) into \( C_6 \) is given in Fig. 10. By Fig.10 and (10), \( r(6) = 6 \). Using this result and (15) we obtain:

\[
r(d) \leq \frac{d-1}{4}, \tag{14}
\]

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 | 6 | 9-10 | 10-11 |

Table 2 illustrates considerable savings in hardware for the proposed space-time signature diagnostics approach over the straightforward diagnostic for \( r(d) \).

For example, 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 \( I_2 = 110,000 \) to \( I_2 = 100,000 \).

Error vectors for \( n=4 \):

\[
(1111), (0111), (0011), (0001)
\]

| Fig. 6a. Line of PEs |

Error vectors for \( n=8 \):

\[
(1111,11), (0111,10), (0011,10), (0001,10)
\]

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

To conclude this section we note that lower bounds (10), (11), upperbounds (12), (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 (8) and (9) are

\[
r(n, d) \geq d \quad \text{and} \quad r(n, d) \geq \lceil \log_2(nd+1) \rceil. \tag{16}
\]

A more specific lower bound can be obtained by the following reasoning. In a rhombic mesh of dimensions \( d \times n \) 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 \( d-1 \).
Error vectors for $h=3$ and $W=5$:

$(11001), (11100), (01110), (00111), (10011)$

$(10001), (11000), (01100), (00110), (00011)$

$(10000), (01000), (00100), (00010), (00001)$

Consider now two "polar zones" in the cube: the vectors of weight $w \geq \lceil \frac{3w}{2} \rceil$ and of weight $w \leq \lfloor \frac{3w}{2} \rfloor$. 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

$$n \leq \sum_{i=1}^{\lfloor \frac{3w}{2} \rfloor} \binom{r}{i} + \sum_{i=\lfloor \frac{3w}{2} \rfloor+1}^{\lceil \frac{3w}{2} \rceil} \binom{r}{i}.$$  \hfill (17)

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

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:

$$H_1 = \begin{pmatrix} I_{d+3} & I_{d+3} & I_{d+3} \\ 00\cdots0 & 00\cdots0 & 11\cdots1 \\ 00\cdots0 & 11\cdots1 & 00\cdots0 \end{pmatrix},$$

$$H_2 = \begin{pmatrix} I_{d+3} & I_{d+3} & I_{d+3} \\ 11\cdots1 & 00\cdots0 & 00\cdots0 \\ 00\cdots0 & 11\cdots1 & 00\cdots0 \end{pmatrix},$$

where $I_{d+3}$ is the $(d+1)$-dimensional identity matrix.

Now let $n = 3(d+1)m$ and $k = \lfloor \log_2 m \rfloor$, where $m = 1, 2, \ldots$. Let $g_0$ be the codeword for the integer $l$ ($0 \leq l \leq 2^k - 1$) in the $k$-bit Gray (reflective) 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
Fig. 8. Recursive Construction for d-Level Binary Tree $T_d$

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

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

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

$$r = k + d + 3 = \left\lfloor \log_2 \frac{n}{3(d + 1)} \right\rfloor + d + 3,$$

(19) provides an upper bound for $r(n, d)$. 
An example of matrix \( H \) for \( n = 18, d = 2 \) is given below:

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

It can be readily shown, that all the syndromes obtained by \( H \) described above are different. Indeed, two errors within the same block \( b_i \) are distinguished by matrices \( H_1 \) or \( H_2 \).

Two errors within different blocks \( b_i \) and \( b_j \) are distinguished by the matrices \( A_1 \) and \( A_2 \), since OI of two consecutive codewords of the Gray code gives always one of these words.

If two errors belong to two disjoint pairs of blocks \( b_{i1} \), \( b_{i2} \) and \( b_{j1} \), \( b_{j2} \), 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 \( b_{i1}, b_{i2} \) and \( b_{j1}, b_{j2} \), respectively. Then their syndromes will differ in the \((d + 2)\)th digit.

Matrices \( H \) for \( n = 2(d + 1) \) may be obtained by slight modifications of the construction given above. We note also that formulas (19) remain valid for \( n = d + 1 \), and \( H = I_{d+1} \).

The lower bounds given by (15) and (16) 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 \( n = 2(d + 1) \):

\[
r(n,2) = \log_2 n + 5 = \log_2 n - 2 \log_2 3 + 5, \quad (20)
\]

\[
r(n,3) = \log_2 n + 6 = \log_2 n - 3 \log_2 3 + 6, \quad (21)
\]

\[
\log_2 n + 6 \leq r(n,4) \leq \log_2 n + 7, \quad (22)
\]

\[
\log_2 n + 7 \leq r(n,3) \leq \log_2 n + 8, \quad (23)
\]

\[
d + 3 \leq r(\delta(d + 1), d) \leq d + 4, \quad (24)
\]

\[
r(2d + 1, d) = d + 3, \quad (25)
\]

\[
r(d + 1, d) = d + 1, \quad (26)
\]

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

The lower bound (2) 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 \( L_1 \approx 10^6 \) equivalent two-input gates, while the suggested method requires only \( L_2 \approx 12 \times 10^5 \) gates.

<table>
<thead>
<tr>
<th>( d = 2 )</th>
<th>( d = 3 )</th>
<th>( d = 4 )</th>
<th>( d = 5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n )</td>
<td>( L )</td>
<td>( U )</td>
<td>( n )</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>20</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>30</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>40</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>60</td>
<td>13</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>70</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>80</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>90</td>
<td>19</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>100</td>
<td>21</td>
<td>21</td>
<td>21</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>( d )</th>
<th>( 2 )</th>
<th>( 3 )</th>
<th>( 4 )</th>
<th>( 5 )</th>
<th>( 6 )</th>
<th>( 7 )</th>
<th>( 8 )</th>
<th>( 9 )</th>
<th>( 10 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n )</td>
<td>L</td>
<td>U</td>
<td>L</td>
<td>U</td>
<td>L</td>
<td>U</td>
<td>L</td>
<td>U</td>
<td>L</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5-5</td>
<td>5-5</td>
<td>5-5</td>
<td>5-5</td>
<td>5-5</td>
<td>5-5</td>
<td>5-5</td>
<td>5-5</td>
<td>5-5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
</tbody>
</table>

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


