Detection and Location of Given Sets of Errors by Nonbinary Linear Codes*

MARK G. KARPOVSKY SAEED M. CHAUDHRY
LEV B. LEVITIN
Research Laboratory of Design and Testing of Computer Hardware
Department of Electrical, Computer and Systems Engineering
Boston University
Boston, Massachusetts 02215, USA

CLAUDIO MORAGA
Department of Computer Science
University of Dortmund
4600 Dortmund 50, Federal Republic of Germany

Abstract—The problem of constructing codes capable of detection and location of a given set of errors is considered. Lower and upper bounds on a number of redundant symbols for an arbitrary set of errors are derived. These codes can be used for error detection and identification of faulty processing elements in multiprocessor systems. To this end, new classes of codes for several types of error sets such as stars, trees and FFTs meshes are presented. The concepts of strong and weak diagnostics (SD and WD, respectively) are introduced and discussed.

Index Terms—Detection and location of a given set of errors, diagnostic of multiprocessor systems or arrays, error detection, error location, linear codes.

1 Introduction

In order for processing elements to cooperate on solving a problem, an interconnection structure (network topology) is provided so that they can communicate with each other. Such multiprocessor systems are increasingly being used for high-performance computing [2], [3], [8], [9], [19]. In many cases, algorithms are developed to exploit specific multiprocessor interconnection structures like

*This work has been supported by the NSF under Grant MIP 9208487 and the NATO under Grant 910411.

meshes, rings, stars, trees, hyper cubes. Some well-known multiprocessor interconnection networks are described in [6], [10], [15].

Built-in self-test and self-diagnostic is becoming a viable alternative for a VLSI chip, board or system design. Techniques based on space and time compression are used to compress output test responses into signatures [1], [20]. Faulty processing elements can be identified by an analysis of distortions in the computed signatures [12]. These approaches aim at reducing the number of signatures required for diagnosis. Such approaches have been used for chip, board and system level testing and diagnosis (see e.g., [18]).

For the diagnosis (fault detection and location) of a system of processing elements the system is represented by a directed graph G whose vertices correspond to processing elements and directed edges correspond to communication links between processing elements [13]. Test data is applied to a set of input vertices of this graph and test responses flow through the graph to a set of output vertices. Due to failures in processing elements or communication links, a set of errors $E(G)$ is observed at the output vertices which corresponds to the structure of the underlying network topology.

In this paper we will consider the problem of constructing error detecting and locating codes for a network of processing elements modeled by a graph. The major problem in the design of error detecting and locating codes is the problem of minimization of the number of check symbols of a code for a given set of errors. In Section 3 we present upper and lower bounds for codes detecting and locating an arbitrary set of errors. Section 4 is devoted to analysis of some important communication topologies for multiprocessor systems and constructions of corresponding codes for detection and location of the underlying graph errors.

2 Definitions and Approach

Let V_q^n denote the n-dimensional vector space over $GF(q)$, $q = 2^b$ ($GF(q)$ is a Galois field of q elements). We denote by supp(v) the support of the n-tuple $v = (v_0, v_1, \ldots, v_{n-1})$, i.e. supp$(v) = \{i \in \{0, 1, \ldots, n-1\} | v_i \neq 0\}$.

Let X denote the set $\{X_0, X_1, \ldots, X_{N-1}\}$ of N processing elements. Consider a digraph G having X as a set of vertices and a set $U = \{U_0, U_1, \ldots, U_{M-1}\}$ of directed edges (b-bit communication links) between vertices of G. Let us denote by $I \subseteq X$ the set of input vertices and by $O \subseteq X$ the set of output vertices. We shall also assume that the graph has no cycles and all output vertices are reachable from at least one input vertex. Let $|O| = n$ ($|O|$ denotes the cardinality of O) and $Y = (y_0, y_1, \ldots, y_{n-1}) \in V_q^n$ be an output vector for the system represented by graph G where $y_i \in GF(q)$ is an output of the corresponding output vertex O_i, $i = 0, 1, \ldots, n-1$ and $q = 2^b$ (for simplicity we assume that all output vertices have b-bit outputs).

The problem to be considered is error detection and location under the assumption of single vertex failures in the graph G. A failure in the graph (system of processing elements) refers to a physical malfunction that cause the undesired event. The effect of a failure is the introduction of errors in the output vector Y. We consider a fault in the graph which alter its output value to $\tilde{Y} = (\tilde{y}_0, \tilde{y}_1, \ldots, \tilde{y}_{n-1})$ where $\tilde{y}_i \in GF(q)$. The error in the graph output Y can be characterized by the error vector $E = (e_0, e_1, \ldots, e_{n-1})$ where $e_i = \tilde{y}_i \oplus y_i$ for $i = 0, 1, \ldots, n-1$ ($\oplus$ is component-wise modulo 2 addition).

Let us first define an error set $E(G)$ characterized by the underlying graph G. In our definition of an error set we assume that at most one vertex or any number of incoming edges to this vertex may fail and a fault in the graph manifest itself by distorting all successor vertices outputs i.e. error propagates along a directed path. This assumption is reasonable for the case when the system of processing elements is tested by a large number of randomly chosen test patterns and, with probability close to one, a distortion will be propagated to all successor vertices [12].

Let $E_j = \{(e_0^{(j)}, e_1^{(j)}, \ldots, e_{n-1}^{(j)})\}$ denote a set of error patterns corresponding to a fault in vertex X_j where $e_i^{(j)} \in \{1, 2, \ldots, q-1\}$ if there exists a directed path from X_j to O_i and $e_i^{(j)} = 0$ otherwise ($e_i^{(j)} = \tilde{y}_i^{(j)} \oplus y_i^{(j)}$, $\tilde{y}_i^{(j)}, y_i^{(j)}$, are faulty and fault free outputs for the output vertex O_i, $i = 0, 1, \ldots, n-1$). The set $E(G) = \bigcup_{j=0}^{N-1} E_j$ of all possible error patterns corresponding to all single vertex failures in G is called the error set for G.

Let $E(G) \subset V_q^n$, $0 \notin E(G)$, be an error set for G. We shall call a linear (n, k), $k \leq n$, code defined by a $(n-k)$ by n parity check matrix H over $GF(q)$ a scheme that allows detection and/or location of error set $E(G)$ where the block length n is equal to the number of output vertices $|O| = n$ in G.

Let $E(G) \subset V_q^n$, $0 \notin E(G)$, be an error set for G. A linear (n, k) block code C over $GF(q)$ of length n defined by a $(n-k)$ by n parity check matrix H

1. detects $E(G)$ if and only if
   (a) for every $E_i \in E(G)$, $HE_i^{tr} \neq 0$ (E_i^{tr} is E_i transposed; all computations are over $GF(q)$).
2. locates $E(G)$ if and only if
   (a) for every $E_i \in E(G)$, $HE_i^{tr} \neq 0$ and
   (b) for every $E_i, E_j \in E(G)$ with $\text{supp}(E_i) \neq \text{supp}(E_j)$, $HE_i^{tr} \neq HE_j^{tr}$.
3. corrects $E(G)$ if and only if
   (a) for every $E_i \in E(G)$, $HE_i^{tr} \neq 0$ and
   (b) for every $E_i, E_j \in E(G)$ with distinct i, j, $HE_i^{tr} \neq HE_j^{tr}$.

A slightly different definition of error locating and correcting code has been given in [17].

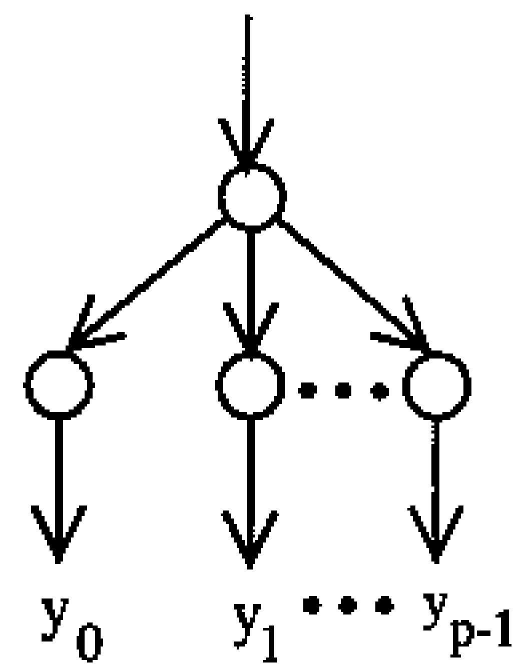

Figure 1: p-ary Star Network Topology.

To illustrate the above definitions let us now consider the problem of fault diagnosis for the p-ary star network topology (see Figure 1). For the p-ary star, the single central processing element (root) is connected to all others, $N = p + 1$, and $n = p$. Due to single vertex or processing element failures we have the following nonzero errors in the p-ary star:

$$E(G) = \left\{ \begin{array}{llll} (e_0, & e_1, & \ldots, & e_{p-1}), \\ (e_0, & 0, & \ldots, & 0), \\ (0, & e_1, & \ldots, & 0), \\ & & \vdots & \\ (0, & 0, & \ldots, & e_{p-1}) \end{array} \right\}, \tag{1}$$

where $e_i \in GF(q) - 0$. Thus, we have $(q-1)^p + p(q-1)$ nonzero error vectors for a p-ary star over $GF(q)$.

For error detection the problem is reduced to optimal construction of a parity check matrix H with n columns and a minimal number of rows $r \leq n$, such that for any error pattern $E_j \in E(G)$, the vector HE_j^{tr} has at least one nonzero component. Let h_i, $1 \leq i \leq r$, denote rows in a parity check matrix H, then to detect all single vertex failures in graph G it is sufficient for any $q > 2$ to have at least one row h_i in H such that $|\text{supp}(h_i) \cap \text{supp}(E_j)| = 1$ for any $E_j \in E(G)$. This condition ensures that any two or more nonzero components in error vectors corresponding to the same vertex failure, E_j, will not compensate and may not produce an all zero syndrome.

For the p-ary star, it is easy to see that to detect any single faulty output vertex, a row of all 1's in H is sufficient, since all error vectors corresponding to a single output vertex failure have only one nonzero component and for the central vertex fault one can take any row with one 1 and $p-1$ 0s. Hence, the following parity check matrix H can be used for detection of p-ary star errors for any $p \geq 2$ and $q \geq 2$:

$$H = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & 0 \end{bmatrix}_{2\times p}. \tag{2}$$

Thus, for any p-ary star, $p \geq 2$, $q \geq 2$, $r = 2$ and we have a class of $(p, p-2)$ star error detecting codes over $GF(q)$.

For identification of faulty vertices the problem is reduced to optimal construction of a parity check matrix H with n columns and a minimal number of rows $r \leq n$, such that for any two error patterns $E_i, E_j \in E(G)$, with different support, $HE_i^{tr} \neq HE_j^{tr}$, where the number of errors with different support is equal to the number of vertices in the graph G.

Since, to locate all single vertex failures in graph G it is necessary that the number of error vectors with different support is equal to the number of vertices N in the graph G we have the following attainable lower bound on a number of outputs n (block length of an (n,k) error locating code) for a single vertex failure locatable graph:

$$n \geq \lceil \log_2(N+1) \rceil . \tag{3}$$

We now present a construction for star error locating codes over $GF(q)$. Let α be a primitive element in $GF(q)$ (α is a primitive element if and only if $\alpha^i \neq \alpha^j$ for $i \neq j$, $i, j = 0, 1, \ldots, q-2$ [16]). The code defined by the following H matrix locates errors in a p-ary star:

$$H = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha & \alpha^2 & \cdots & \alpha^{p-1} \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \end{bmatrix}_{4\times p} , \tag{4}$$

where $4 \leq p < q$.

For an output vector $Y = (y_0, y_1, \ldots, y_{n-1})$, we define its syndrome $S(Y) = (S_1, S_2, \ldots, S_{r=n-k})$ as

$$S(Y) = HY^{tr}. \tag{5}$$

Diagnosis of a single vertex failure will consists of two steps: syndrome computation and association of the syndromes to a faulty vertex. A straightforward approach to the syndrome computation is via a combinational logic circuit that implements the parity check matrix and the association of the syndrome to a faulty vertex can be specified by a location algorithm.

The syndrome computation for the parity check matrix described in (4) can be implemented with $2(p-1)$ $GF(q)$ adders and $(p-1)$ multipliers (note that here we need a multiplier that multiplies a field element from $GF(q)$ by a fixed element from the field).

The location algorithm for the above $(p, p-4)$ p-ary star error locating codes is described as follows:

Let S_i, $i = 1, 2, 3, 4$ denote the syndromes obtained.

1. If $S_i = 0$, $i = 1, 2, 3, 4$: no error, end.

2. If $S_3 \neq 0$ and $S_4 \neq 0$: error in the central vertex (root), end.

3. If $S_3 = 0$ or $S_4 = 0$: error in vertex (leaf) j, $0 \leq j \leq p-1$, where $\alpha^j = S_2/S_1$.

4. End. □

For the case of a p-ary star with $p = 2^i - 2$ and $q = 2$ one may choose as a check matrix for error location a matrix with $2^i - 2$ different nonzero i-tuples as its columns except for the all 1 vector. For example the following check matrix can locate all single vertex failures in the star with $p = 2^3 - 2 = 6$ and $q = 2$:

$$H = \begin{bmatrix} 000111 \\ 011001 \\ 101010 \end{bmatrix}. \tag{6}$$

Therefore, for the p-ary star with $p = 2^i - 2$:

$$r = i, \tag{7}$$

and we have a class of $(2^i - 2, 2^i - 2 - i)$ perfect star error locating codes over $GF(2)$ [11].

It is interesting to note that the parity check matrix for star error detecting codes (2) is over $\{0, 1\} \subseteq GF(q)$. Such codes are particularly simple to implement since the check symbols are obtained using only additions in $GF(q)$, no multiplications are needed. Therefore, from the viewpoint of hardware implementation it is advantageous to have codes with parity check matrix over $\{0, 1\}$ (see e.g., [12]).

In the next section, we present upper and lower bounds on the minimum number of check symbols for codes detecting and locating an arbitrary set of errors.

3 Error-Detection and Location Capabilities of Linear Codes

A lower bound on $r = n - k$, minimum number of redundant symbols, in any (n, k) block code capable of detecting an error set $E(G)$ can be proved as follows. Let $E(G) \subset V_q^n$, $0 \notin E(G)$, be the set of errors we wish to detect. A linear (n, k) code over $GF(q)$ has q^k code vectors. If this code is to detect the set $E(G)$ of errors, all error vectors must not be in the code. Thus the number of code vectors must be no greater than the total number of vectors in the space minus the number of error vectors in $E(G)$:

$$q^k \leq q^n - |E(G)|, \tag{8}$$

where $|E(G)|$ is the cardinality of $E(G)$. Thus

$$r \geq n - \lfloor \log_q(q^n - |E(G)|) \rfloor . \tag{9}$$

Another and more efficient lower bound on a number of redundant symbols for a code detecting a set of errors $E(G)$ can be proved as follows. Consider a graph $G(E) = (E(G), U)$ having the error set $E(G)$, $0 \notin E(G)$, as a set of vertices and U a set of edges $\{(E_i, E_j) | E_i \oplus E_j \in E(G)\}$. Let $E_i \oplus E_j = E_k \in E(G)$ and let H be a check matrix for a code detecting the set of errors $E(G)$. Since H is a linear code, $H(E_i^{tr} \oplus E_j^{tr}) = HE_i^{tr} \oplus HE_j^{tr} = HE_k^{tr} \neq 0$. Therefore, $HE_i^{tr} \neq HE_j^{tr}$, which means that the syndromes for two errors E_i and E_j must be different if E_i is connected to E_j in $G(E)$.

Let $\gamma(E)$ denote the chromatic number for $G(E)$ ($\gamma(E)$ is a minimal number of colors required to color vertices of $G(E)$ in such a way that no two neighboring vertices have the same color; techniques for graph coloring with lower and upper bounds for $\gamma(E)$ can be found e.g. in [4]. Then we have the following lower bound on a minimal number r of check symbols in a code detecting $E(G)$:

$$r \geq \lceil \log_q (\gamma(E) + 1) \rceil . \tag{10}$$

We note that the above lower bound is attainable. For example, for a p-ary star (see Figure 1), $p > 2$, and $q = 2$:

$$H = \begin{cases} [11\ldots1]_{1\times p} & p = \text{odd}, \\ \begin{bmatrix} 11\ldots1 \\ 10\ldots0 \end{bmatrix}_{2\times p} & p = \text{even}. \end{cases} \tag{11}$$

We also note that for the classical case when $E(G) = \{e | 0 < ||e|| \leq 2t\}$ ($||e||$ denote the number of nonzero components in e) we have $\gamma(E) = \sum_{i=1}^{t} \binom{n}{i}$ and (10) is the well known Hamming bound [16].

An upper bound on a number r of redundant symbols for a code detecting a set of errors $E(G)$ for any $n, k < n$, and any $E(G) \subset V_q^n$, $0 \notin E(G)$, is

$$r \leq \lceil \log_q (|E(G)|(q-1) + 1) \rceil . \tag{12}$$

This bound follows from the fact that a linear (n, k) code over $GF(q)$ with q^k code vectors detects error set $E(G)$ if and only if all code vectors are in the set $V_q^n - E(G)$ and in any set with $q^n - 1 - |E(G)| \geq q^n - (q^{n-k+1} - 1)(q-1)^{-1}$ nonzero vectors, there exists a linear subspace with q^k vectors [14].

Lower and upper bounds for a code locating an error set $E(G)$ can be obtained from the above bounds for error detection by replacing error set $E(G)$ with $E(G) \cup \{E_i \oplus E_j | \text{supp}(E_i) \neq \text{supp}(E_j), E_i, E_j \in E(G)\}$. This is due to the fact that if a code defined by the parity check matrix H detects a set $E(G) \cup \{E_i \oplus E_j | \text{supp}(E_i) \neq \text{supp}(E_j), E_i, E_j \in E(G)\}$, then for any $E_i, E_j \in E(G)$ with $\text{supp}(E_i) \neq \text{supp}(E_j)$, $HE_i^{tr} \neq HE_j^{tr}$ i.e., code locates error set $E(G)$.

Since all nonzero syndromes must be different for all single vertex failures the following attainable bounds on a minimum number of check symbols required for a (n, k) code over $GF(q)$ locating error set $E(G)$ hold:

$$\lceil \log_q(N+1) \rceil \leq r \leq n. \tag{13}$$

We note also that a linear (n, k) code with a parity check matrix H locating an error set $E(G)$, $0 \notin E(G)$ corrects $E(G)$ if and only if for every distinct $E_u, E_v \in E(G)$ with $\mathrm{supp}(E_u) = \mathrm{supp}(E_v)$ there exists a pair $E_i, E_j \in E(G) \cup 0$ with $\mathrm{supp}(E_i) \neq \mathrm{supp}(E_j)$ such that $E_u \oplus E_v = E_i \oplus E_j$. To show this assume that a code with check matrix H locates error set $E(G)$ but does not correct $E(G)$. Therefore, there exist at least one pair E_u, E_v such that $HE_u^{tr} = HE_v^{tr}$ where E_u, E_v have the same support. Since there exist at least one pair E_i, E_j such that $E_u \oplus E_v = E_i \oplus E_j$ and E_i, E_j have different support, we have $H(E_u^{tr} \oplus E_v^{tr}) = 0$ which implies $HE_i^{tr} = HE_j^{tr}$ which is a contradiction, because the code locates $E(G)$. We note that the above condition is a necessary and sufficient condition on the error set $E(G)$ such that if and only if this condition is satisfied any linear code locating $E(G)$ will also correct $E(G)$.

For example, for p-ary star with $p = 5$ and $q = 2^2$, $E(G) = \{(e_0, e_1, e_2, e_3, e_4)$, $(e_0, 0, 0, 0, 0)$, $(0, e_1, 0, 0, 0)$, $(0, 0, e_2, 0, 0)$, $(0, 0, 0, e_3, 0)$, $(0, 0, 0, 0, e_4\}$ where $e_i \in \{1 = 01, \alpha = 10, \alpha^2 = 11\}$. A code locating $E(G)$ does not guarantee error correction since for two errors $E_u = (1, 1, 1, 1, 1)$, $E_v = (1, 1, \alpha, \alpha, \alpha)$ there is no pair E_i, E_j with different support such that $E_u \oplus E_v = E_i \oplus E_j = (0, 0, \alpha^2, \alpha^2, \alpha^2)$ (Note $\alpha^2 + \alpha + 1 = 0$).

Using the above arguments one can see that any code over $GF(q)$ locating up to t independent errors ($E(G) = \{e | 0 < ||e|| \leq 2t\}$) can also correct t errors. The same is also true for codes locating burst errors.

4 Codes for Diagnosis of Multiprocessor Systems

In Section 2 we have shown that the problem of hardware minimization for diagnosis of a system of processing elements modeled by a graph can be reduced to the design of a code with a minimal number of check symbols detecting and locating graph errors. In this section we will present several nearly optimal constructions for codes detecting and locating errors in tree and Fast Fourier Transform (FFT) interconnection networks. These interconnection networks have been widely used (see e.g. [5], [6] and [10]).

4.1 Detection and Location of Tree Errors

Let T_h be a p-ary full tree of height h ($p \geq 2, h \geq 2$) (see Figure 2). The height h is the length of a longest path from the root to any leaf. Here we assume that input vertex is the root and output vertices are $n = p^{h-1}$ leaves of the tree.

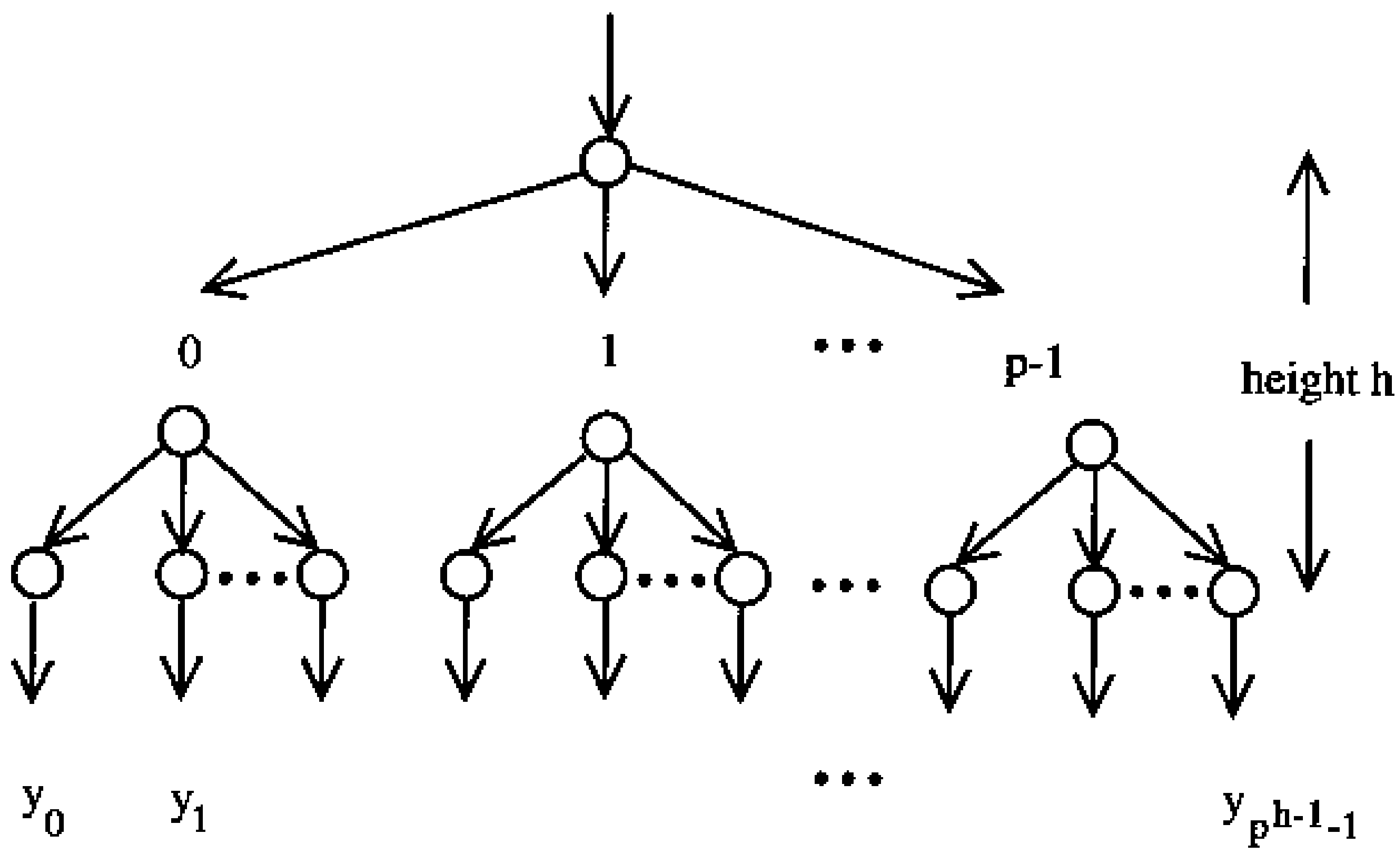

Figure 2: p-ary Full Tree Network Topology.

For a p-ary tree of height h, the set of errors $E(G)$ is:

$$\left\{ \begin{array}{llllllllll}
(e_0, & e_1, & \ldots, & e_{p^{h-2}-1}, & e_{p^{h-2}}, & \ldots, & e_{2p^{h-2}-1}, & e_{(p-1)p^{h-2}}, & \ldots, & e_{p^{h-1}-1}), \\
(e_0, & e_1, & \ldots, & e_{p^{h-2}-1}, & 0, & \ldots, & 0, & 0, & \ldots, & 0), \\
(0, & 0, & \ldots, & 0, & e_{p^{h-2}}, & \ldots, & e_{2p^{h-2}-1}, & 0, & \ldots, & 0), \\
 & & & & & & & & \vdots & \\
(0, & 0, & \ldots, & 0, & 0, & \ldots, & 0, & e_{(p-1)p^{h-2}}, & \ldots, & e_{p^{h-1}-1}), \\
 & & & & & & & & \vdots & \\
(e_0, & 0, & \ldots, & 0, & 0, & \ldots, & 0, & 0, & \ldots, & 0), \\
(0, & e_1, & \ldots, & 0, & 0, & \ldots, & 0, & 0, & \ldots, & 0), \\
 & & & & & & & & \vdots & \\
(0, & 0, & \ldots, & 0, & 0, & \ldots, & 0, & 0, & \ldots, & e_{p^{h-1}-1})
\end{array} \right\}, \qquad (14)$$

where $e_i \in GF(q) - 0$ and $|E(G)| = \sum_{i=0}^{h-1} p^i (q-1)^{p^{h-1-i}}$.

The recursive construction for check matrices of $(p^{h-1}, p^{h-1} - h)$, $p \geq 2$, $q > 2$, tree error detecting codes is given by:

$$H_h = \left[\begin{array}{c} \overbrace{H_{h-1} H_{h-1} \cdots H_{h-1}}^{p} \\ W \end{array} \right], \qquad (15)$$

where W is a row vector of one 1 followed by $p^{h-1} - 1$ 0's and

$$H_2 = \left[\begin{array}{c} 11 \cdots 1 \\ 10 \cdots 0 \end{array} \right]_{2 \times p}, \qquad (16)$$

(H_h is a check matrix for the p-ary tree of height h).

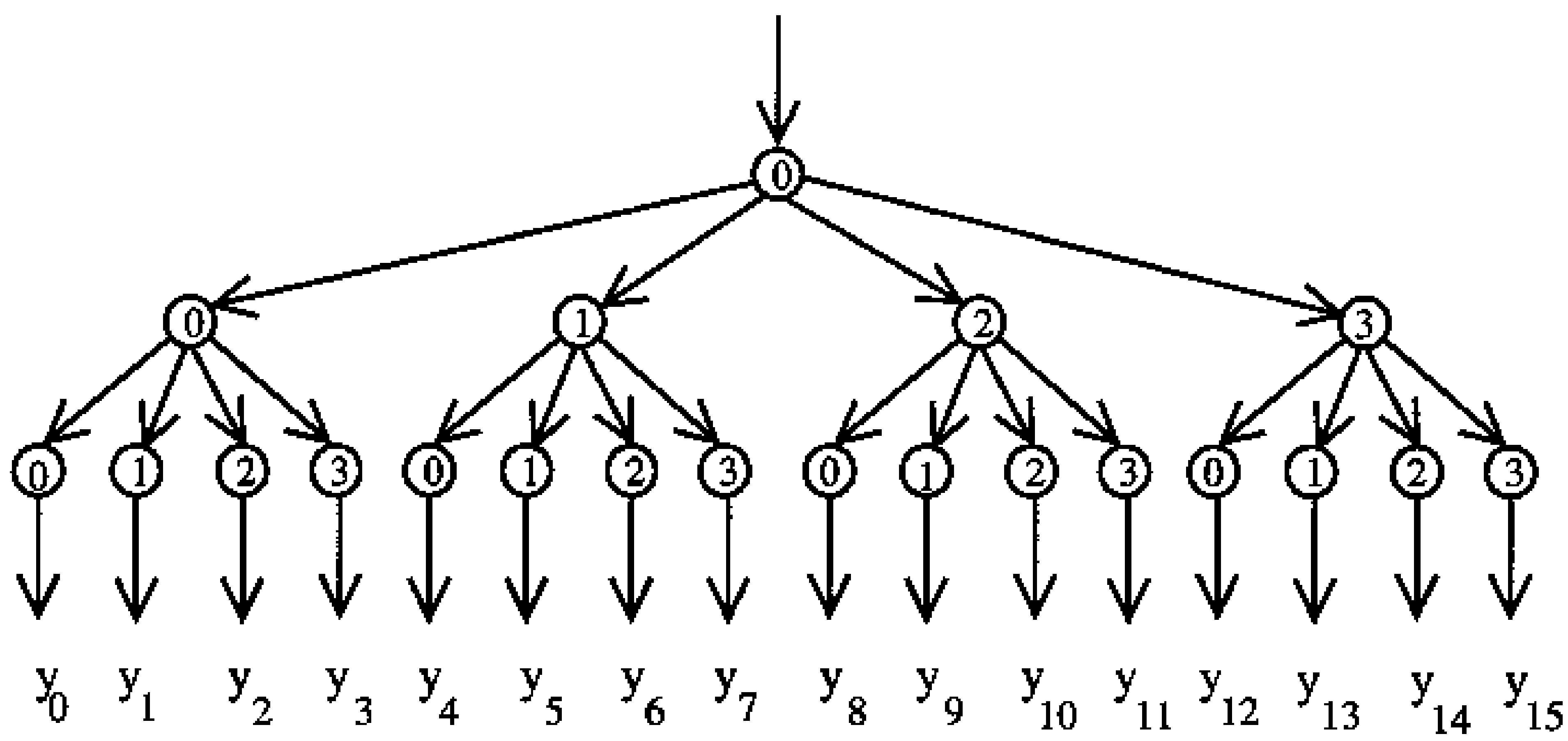

Figure 3: 4-ary Full Tree of Height $h = 3$ — Example.

It can be easily shown that all syndromes obtained by H_h for tree errors are not equal to zero and the number of rows r in H_h is equal to the height h of the tree. Thus we have the class of $(p^{h-1}, p^{h-1} - h)$ p-ary tree error detecting codes.

The complexity L for the syndrome computing network in terms of a number of $GF(q)$ adders is

$$L = (p^{h-1} - 1)L_{\oplus}, \tag{17}$$

where $L_{\oplus}$ is the complexity of a $GF(q)$ adder and H_h given by (15) is optimal from the point of view of decoding complexity (note that elements of H_h are 0 or 1 for any q, therefore no multipliers are required).

Example. Consider the 4-ary full tree of height $h = 3$ over $GF(2^3)$ shown in Figure 3. This tree has $|E(G)| = (8-1)^{16} + 4(8-1)^4 + 16(8-1) \simeq 3.32 \times 10^{13}$ different error patterns due to 21 different single vertex faults. These errors are:

$$\left\{\begin{array}{cccccccccccccccc}
(e_0, & e_1, & e_2, & e_3, & e_4, & e_5, & e_6, & e_7, & e_8, & e_9, & e_{10}, & e_{11}, & e_{12}, & e_{13}, & e_{14}, & e_{15}), \\
(e_0, & e_1, & e_2, & e_3, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0), \\
(0, & 0, & 0, & 0, & e_4, & e_5, & e_6, & e_7, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0), \\
(0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & e_8, & e_9, & e_{10}, & e_{11}, & 0, & 0, & 0, & 0), \\
(0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0 & e_{12}, & e_{13}, & e_{14}, & e_{15}), \\
(e_0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0), \\
(0, & e_1, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0), \\
 & \vdots & & & & & & & & & & & & & & \\
(0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & e_{15}),
\end{array}\right\}, \tag{18}$$

where $e_i \in GF(2^3) - 0$.

Based on the construction given in (15) we have the following parity check

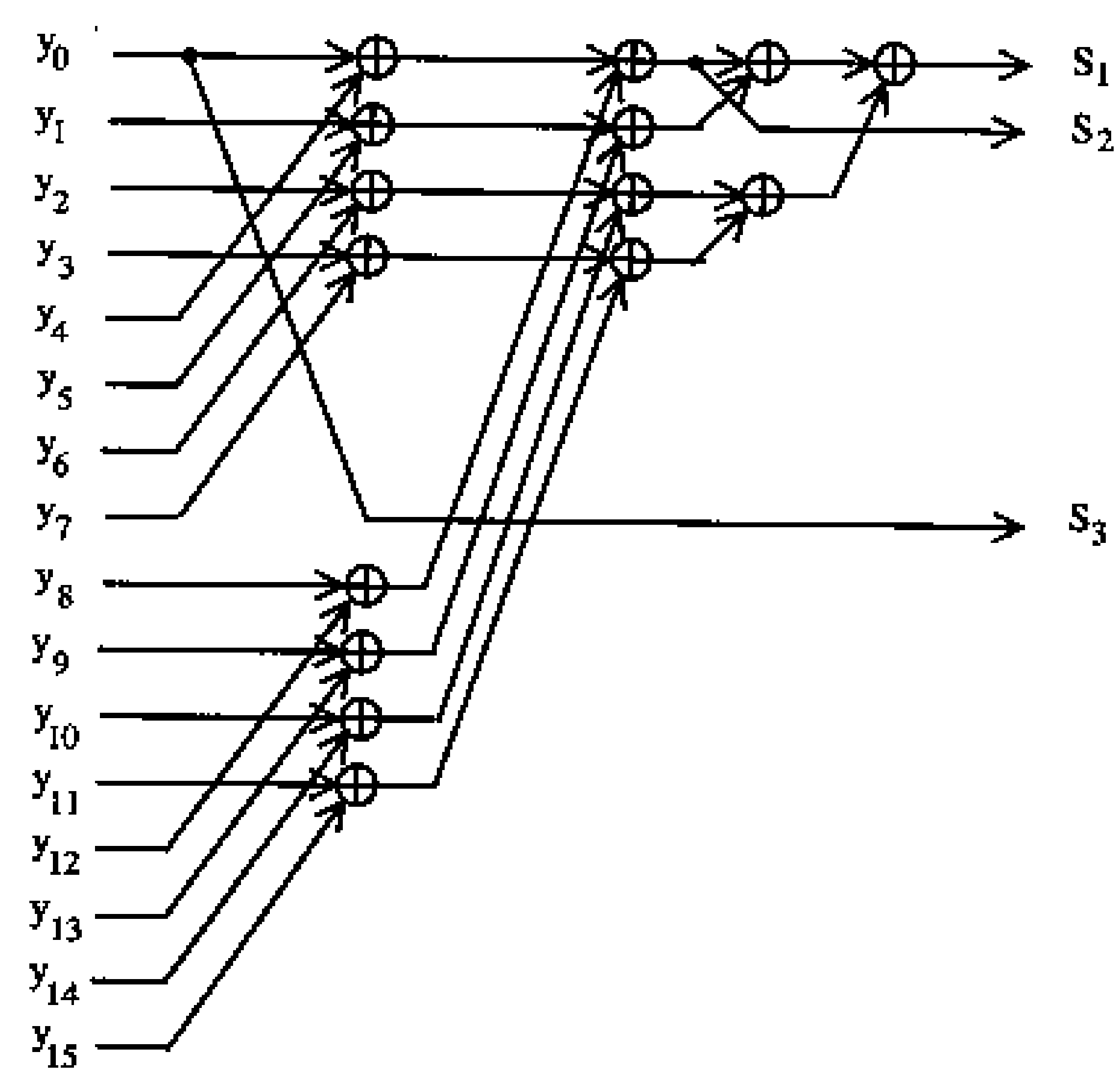

Figure 4: Syndrome Computing Network for the 4-ary (16, 13) Tree Error Detecting Code — Example.

matrix for the (16, 13) tree error detecting code over $GF(2^3)$

$$H_3 = \begin{bmatrix} 1111 & 1111 & 1111 & 1111 \\ 1000 & 1000 & 1000 & 1000 \\ 1000 & 0000 & 0000 & 0000 \end{bmatrix}, \tag{19}$$

and the combinational network for computing $S(Y) = (S_1, S_2, S_3) = H_3 Y^{tr}$ is shown in Figure 4. □

For the case of a p-ary full tree over $GF(2)$ of any height h, $h \geq 2$, we have

$$r = \begin{cases} 1, & p = \text{odd}, \\ 2, & p = \text{even}. \end{cases} \tag{20}$$

A recursive construction for check matrices for the class of $(2^{h-1}, 2^{h-1} - 2)$ tree error detecting codes over $GF(2)$ can be obtained in the following way. We define the following mapping for columns in check matrix H_h:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \mapsto \begin{bmatrix} 00 \\ 00 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} 11 \\ 01 \end{bmatrix},$$
$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \mapsto \begin{bmatrix} 10 \\ 11 \end{bmatrix}, \quad \begin{bmatrix} 1 \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} 01 \\ 10 \end{bmatrix}. \tag{21}$$

The parity check matrix H_h can be obtained recursively from H_{h-1} by (21) where $H_2 = I_2$ (I_2 is an identity matrix of 2 by 2). For example, the matrix H_4 for the binary full tree of height four, $h = 4$, $n = 8$, is given by:

$$H_4 = \begin{bmatrix} 01 & 11 & 10 & 01 \\ 10 & 01 & 11 & 10 \end{bmatrix}. \tag{22}$$

We now present a construction for the class $(p^{h-1}, p^{h-1} - 3h + 2)$ tree error locating codes. The code defined by the following recursive definition of H_h locates all tree errors:

$$H_h = \begin{bmatrix} H_{h-1} & H_{h-1} & \cdots & & H_{h-1} & & \\ 11\cdots1 & \alpha\alpha\cdots\alpha & \cdots & \alpha^{p-1} & \alpha^{p-1} & \cdots & \alpha^{p-1} \\ 10\cdots0 & 00\cdots0 & \cdots & 0 & 0 & \cdots & 0 \\ 00\cdots0 & 10\cdots0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix}, \tag{23}$$

where

$$H_2 = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha & \alpha^2 & \cdots & \alpha^{p-1} \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \end{bmatrix}_{4\times p}, \tag{24}$$

$4 \le p < q$ and α is primitive in $GF(q)$. Thus for any p-ary, $4 \le p < q$, tree $r = 3h - 2$ and we have the class of $(p^{h-1}, p^{h-1} - 3h + 2)$ p-ary tree error locating codes over $GF(q)$. For the case of $p = 3$ one can also use the above construction (23) with $H_2 = I_3$ (I_3 is an identity matrix of 3 by 3). Therefore, for a p-ary tree with $p = 3$, $q > p$, $r = 3h - 3$.

For binary tree over $GF(q)$ the following recursive construction can be used for the class of $(2^{h-1}, 2^{h-1} - 2h + 2)$ binary tree error locating codes over $GF(q)$:

$$H_h = \begin{bmatrix} H_{h-1} & H_{h-1} \\ 10\cdots0 & 00\cdots0 \\ 00\cdots0 & 10\cdots0 \end{bmatrix}, \tag{25}$$

where $H_2 = I_2$. Thus, for a binary tree over $GF(q)$ of height h, $r = 2h - 2$ and we have the class of $(2^{h-1}, 2^{h-1} - 2h + 2)$ binary tree error locating codes over $GF(q)$.

The complexity L for the syndrome computing network in terms of numbers of $GF(q)$ adders and multipliers is:

$$L = (2(p^{h-1} - p) + h(p - 1))L_{\oplus} + ((h - 1)(p - 1))L_{\otimes}, \tag{26}$$

where $L_{\otimes}$ is a complexity of a multiplier that multiplies a field element from $GF(q)$ by a fixed element from the same field.

The decoding procedure for tree codes is very simple. Let us denote the syndromes obtained as

$$S(Y) = H_h Y^{tr} = \begin{bmatrix} S^{h-1} \\ S_1^h \\ S_2^h \\ S_3^h \end{bmatrix}, \qquad (27)$$

where S^{h-1} are syndromes due to the $[H_{h-1} H_{h-1} \cdots H_{h-1}]$ part of H_h (see (23)) and S_1^h, S_2^h, S_3^h are syndromes for the last three rows of the parity check matrix H_h. Let S^1 denotes the syndrome for the all 1 row. The location algorithm to find a faulty vertex is described as follows:

1. If $S_i = 0, i = 1, 2, \ldots, 3h - 2$: no error, end.
2. Let $j = h$.
3. If both $S_2^j \neq 0$ and $S_3^j \neq 0$: error location is the root of the tree of height j, end.
4. For $j > 2$, if either $S_2^j = 0$ or $S_3^j = 0$: error location is in the subtree k, $0 \leq k \leq p-1$, where $\alpha^k = S_1^j / S_2^{j-1}$; for $j = 2$, if either $S_2^2 = 0$ or $S_3^2 = 0$: error location is in vertex (leaf) k, $0 \leq k \leq p-1$, where $\alpha^k = S_1^2 / S^1$: end.
5. Repeat steps 3 and 4 for tree of height $j = j - 1$.
6. End.

Example continued: The parity check matrix H_3 for the 4-ary (16, 9) tree error locating code is:

$$H_3 = \left[\begin{array}{cccc|cccc|cccccccc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & \alpha & \alpha^2 & \alpha^3 & 1 & \alpha & \alpha^2 & \alpha^3 & 1 & \alpha & \alpha^2 & \alpha^3 & 1 & \alpha & \alpha^2 & \alpha^3 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & \alpha & \alpha & \alpha & \alpha & \alpha^2 & \alpha^2 & \alpha^2 & \alpha^2 & \alpha^3 & \alpha^3 & \alpha^3 & \alpha^3 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right]. \qquad (28)$$

Suppose that root of subtree 1 (see Figure 3) is faulty and the received message is:

$$E = \tilde{Y} \oplus Y = (0, 0, 0, 0, 1, \alpha, 1, \alpha^3, 0, 0, 0, 0, 0, 0, 0, 0). \qquad (29)$$

Then the syndromes of this message are

$$S(E) = HE^{tr} = (\alpha, \alpha^5, \alpha, \alpha, \alpha^2, 0, \alpha)^{tr}. \qquad (30)$$

This yields $S_2^3 = 0$, therefore the error is in the subtree 1 since, $S_1^3 / S_2^2 = \alpha^2 / \alpha = \alpha$, $i = 1$. Since $S_2^2 \neq 0$ and $S_3^2 \neq 0$, error is in the root of subtree 1 of height 2. The combinational network for computing $S(Y) = H_3 Y^{tr}$ is shown in Figure 5.

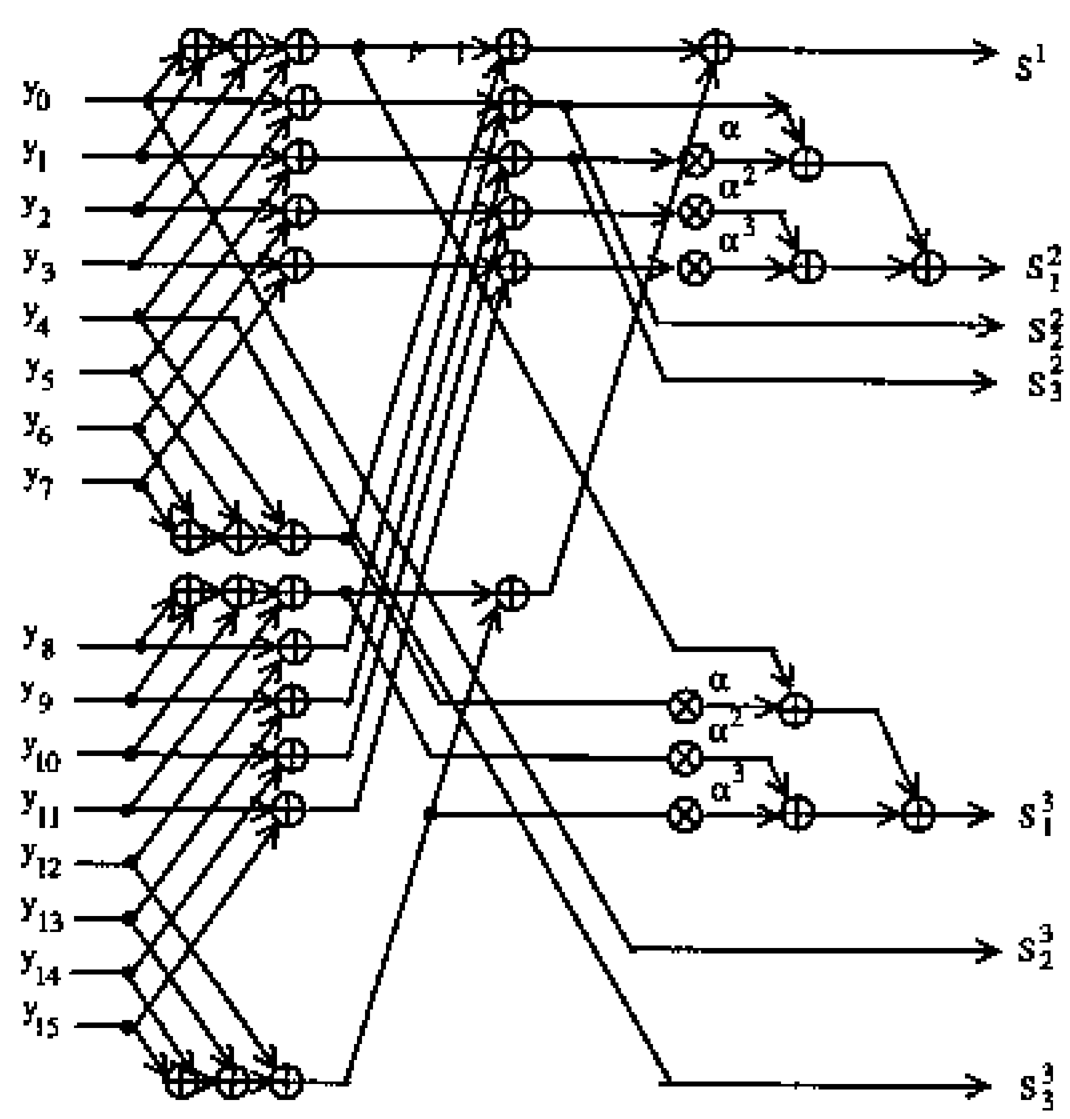

Figure 5: Syndrome Computing Network for the 4-ary (16, 9) Tree Error-Locating Code — Example.

4.2 Detection of Errors in Fast Fourier Transform (FFT) Networks

The results presented above for detection of tree errors can be extended to other graphs of practical interest containing tree structures as subgraphs (i.e. any single vertex failure propagates through the graph in a tree-like manner). Below we consider an important application of tree-like codes for detection of errors in Fast Fourier Transform (FFT) network [5].

For n-point FFT, there are $N = n \log_2 n$ vertices interconnected with $\log_2 n$ levels of butterfly structures, e.g., the graph for the 8-point FFT (decimation-in-frequency (DIF)) is shown in Figure 6.

If we also consider input fanout branches as possible source of errors, there are $n(\log_2 n{+}1)$ single faults in the n-point FFT graph. Due to these single faults

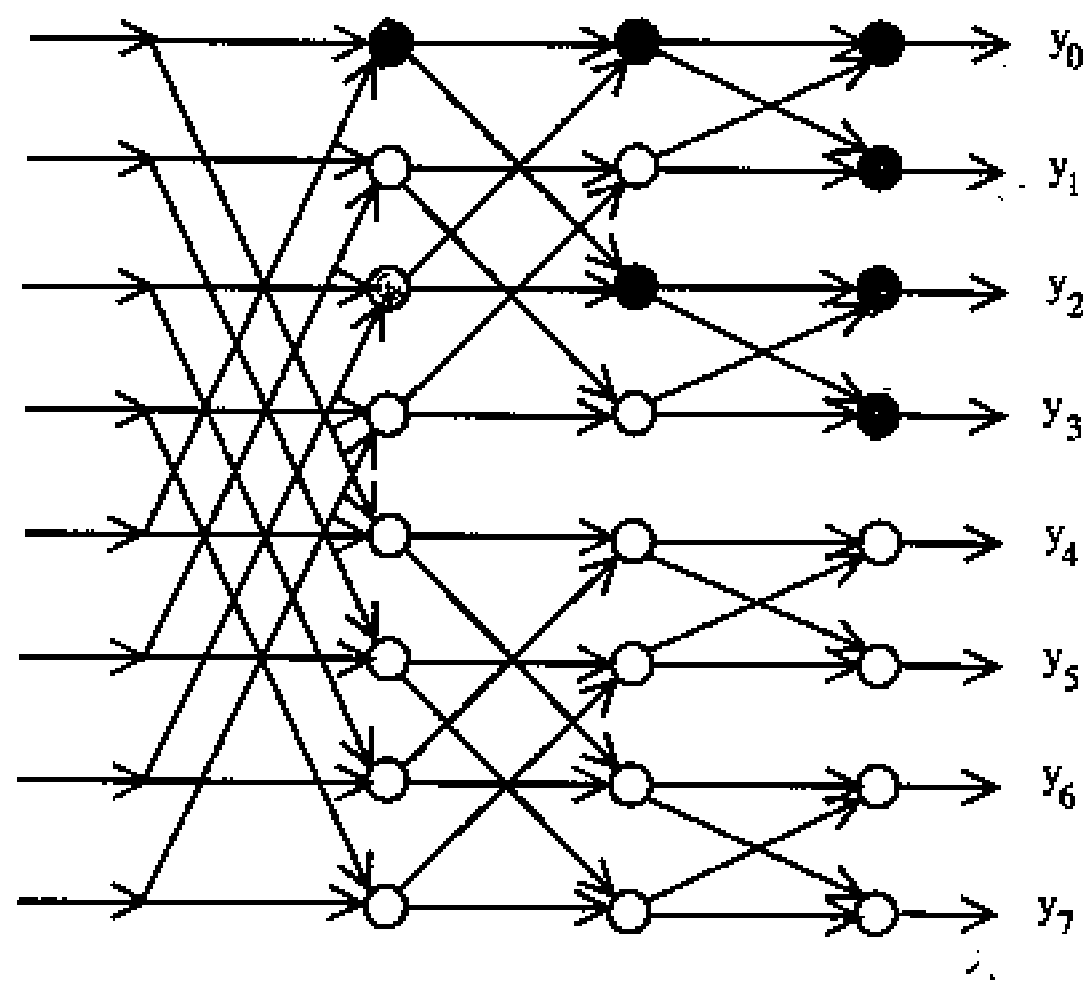

Figure 6: Eight-Point DIF FFT Network Topology.

we have the following nonzero errors for the 8-point FFT graph of Figure 6:

$$E(G) = \left\{ \begin{array}{llllllll} (e_0, & e_1, & e_2, & e_3, & e_4, & e_5, & e_6, & e_7), \\ (e_0, & e_1, & e_2, & e_3, & 0, & 0, & 0, & 0), \\ (0, & 0, & 0, & 0, & e_4, & e_5, & e_6, & e_7), \\ (e_0, & e_1, & 0, & 0, & 0, & 0, & 0, & 0), \\ (0, & 0, & e_2, & e_3, & 0, & 0, & 0, & 0), \\ (0, & 0, & 0, & 0, & e_4, & e_5, & 0 & 0), \\ (0, & 0, & 0, & 0, & 0 & 0 & e_6, & e_7), \\ (e_0, & 0, & 0, & 0, & 0, & 0, & 0, & 0), \\ (0, & e_1, & 0, & 0, & 0, & 0, & 0, & 0), \\ & \vdots & & & & & & \\ (0, & 0, & 0, & 0, & 0, & 0, & 0, & e_7), \end{array} \right\}, \tag{31}$$

where $e_i \in GF(q) - 0$ and $|E(G)| = \sum_{i=0}^{p} 2^i (q-1)^{2^{p-i}}$.

A recursive construction for the $(2^p, 2^p - p - 1)$ DIF FFT error detecting codes is:

$$H_{2^p} = \left[\begin{array}{c} H_{2^{p-1}} H_{2^{p-1}} \\ W \end{array} \right], \tag{32}$$

where W be a row vector of one 1 followed by $2^p - 1$ 0's and

$$H_2 = \left[\begin{array}{c} 11 \\ 10 \end{array} \right]. \tag{33}$$

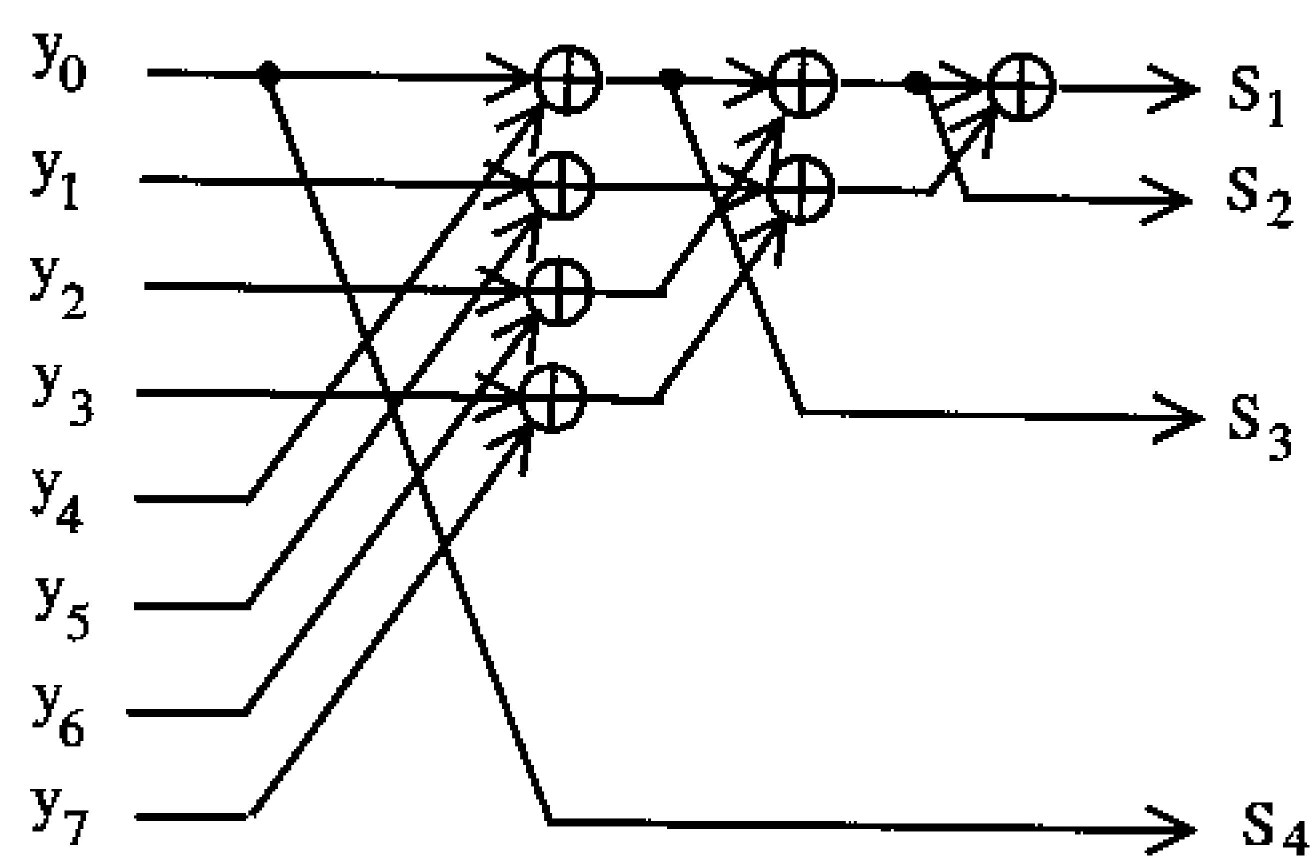

Figure 7: Syndrome Computing Network for the $(8,4)$ DIF FFT Error Detecting Code.

We note that a similar construction for the class of $(2^p, 2^p - p - 1)$, decimation in time (DIT) FFT error detecting codes is also possible.

Based on the above construction we have the following parity check matrix H_{2^3} for the $(8,4)$ DIF FFT error detecting code over $GF(q)$:

$$H_{2^3} = \begin{bmatrix} 11 & 11 & 11 & 11 \\ 10 & 10 & 10 & 10 \\ 10 & 00 & 10 & 00 \\ 10 & 00 & 00 & 00 \end{bmatrix}. \tag{34}$$

The complexity L for the syndrome computing network is:

$$L = (2^p - 1)L_{\oplus}, \tag{35}$$

and H_{2^p} given by (32) is optimal from the point of view of decoding complexity (note that elements of H_{2^p} are 0 or 1 for any q, therefore no multipliers are required). The combinational network for computing $S(Y) = (S_1, \ldots, S_4) = H_{2^3}Y^{tr}$ is shown in Figure 7.

We note that for FFT graphs error location is not possible due the fact that different faults may have the same error patterns.

In Table 1 we summarize our results on linear error detecting and locating codes for different meshes (stars, binary and non binary full trees and FFTs).

Table 1: Minimal Numbers of Check Symbols for Strong Diagnosis of Multiprocessor Interconnection Networks.

Graph	Parameters N, n	Number of Check Symbols r	
		Error Detection	Error Location
p-ary star	$p+1, p$	2^*, $p \geq 2$, $q > 2$ 1, p odd, $q = 2$ 2, p even, $q = 2$	4, $4 \leq p < q$ i, $p = 2^i - 2$, $q = 2$
Binary full tree of height h	$2^h - 1, 2^{h-1}$	h^*, $h \geq 2$, $q > 2$ 2, $q = 2$	$2h - 2$, $h \geq 2$, $q > 2$
p-ary full tree of height h	$\frac{p^h - 1}{p-1}, p^{h-1}$	h^*, $p \geq 2$, $q > 2$ 1, p odd, $q = 2$ 2, p even, $q = 2$	$3h - 2$, $4 \leq p < q$ $3h - 3$, $p = 3$, $q > p$
2^p point FFT	$p2^p, 2^p$	$p + 1^*$	—

*Parity Check Matrix is over $\{0, 1\}$.

5 Weak Diagnostics of Multiprocessor Systems

The approach to error location adopted in the previous sections is based on rather a strong requirement that for every two errors, E_i, E_j if $\mathrm{supp}(E_i) \neq \mathrm{supp}(E_j)$, then $HE_i^{tr} \neq HE_j^{tr}$ (cf. Section 2).

This requirement can be reformulated in the following way. Denote by $U_i = \{E_{ik}\}$ the set of all errors E_{ik} with the same support $\mathrm{supp}(E_{ik}) = Y_i \subseteq Y$, such that $U_i \subseteq E(G)$. Then, for any two U_i, U_j, $i \neq j$,

$$H(U_i) \cap H(U_j) = \phi. \tag{36}$$

(Note that the check matrix H is a linear operator $H : V_q^n \to V_q^r$).

Condition (36) defines what can be called strong diagnostics (SD) of a given class of errors $E(G)$ in G. However this requirement may be too restrictive and, in some cases, may result in too large redundancy r. Moreover, it should be born in mind, that a fault that can manifests itself as an error with a largest possible support Y_i can, in fact, manifests itself as an error with a smaller support $Y_i' \subset Y_i$. If $|Y_i| = l$, then the fraction of such cases is of order l/q. Since in SD we assume that any fault manifest itself as an error with the maximum possible support (any output which can go wrong does so) we can not guarantee correct location of that fraction of faults.

It looks attractive, therefore, to consider a different approach which would relax the requirement (36), thereby considerably decreasing the value of r, and, on the other hand, would keep the fractions of faults denying location reasonably small for large q (of $\Theta(1/q)$).

Denote $V_i = \{E_{ik} | \text{supp}(E_{ik}) \subseteq Y_i\}$ the set of all errors whose support are subsets of Y_i. Obviously, V_i is an m-dimensional coordinate subspace of V_q^n, where $m = |Y_i|$. Then the problem of error location can be formulated as follows: For a given set $\{V_i\}$ find a linear mapping $H : V_q^n \to V_q^r$ such that for any $i \neq j$,

$$H(V_i) \neq H(V_j). \tag{37}$$

We will call it weak diagnostics (WD).

The linear operator H can be viewed as a check matrix of a linear code of length n and dimension $(n-r)$ over $GF(q)$ which is a sub space of V_q^n and the kernel of the operator H.

Below we consider a few typical system topologies, give the explicit constructions for H and estimate the probability of correct fault location.

5.1 Faults in Stars

A star is a tree of height 2. Consider a p-ary star, where $p < q$. Here $n = p$ is the number of leaves. Let α be a primitive element in $GF(q)$ and consider a $(2 \times n)$ matrix

$$H_2 = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \end{bmatrix}. \tag{38}$$

Let (S_1, S_2) be the syndrome. Then the ratio S_2/S_1 reveals the faulty leaf:

$$S_2/S_1 = \alpha^i, \quad i = 0, 1, \ldots, n-1 \tag{39}$$

The misdiagnosis occurs if (39) holds, but in fact the root is faulty. For every i, (39) defines a subspace in V_q^n of dimension $(n-1)$. The intersection of all those subspaces corresponds to the case $S_1 = 0$, $S_2 = 0$, which defines a subspace of dimension $(n-2)$. Thus the fraction of misdiagnosis cases is

$$\omega_e = \frac{nq^{n-1} - q^{n-2}}{q^n} = \frac{nq-1}{q^2} \approx \frac{n}{q}. \tag{40}$$

In a practical situation, $n \leq 100$, $q = 2^{32}$, and $\omega_e \approx 2.5 \times 10^{-8}$. This negligible probability of mislocation allows us to reduce the redundancy by a factor of 2: $r = 2$, instead of $r = 4$ for SD (cf. Section 4.1).

5.2 Faults in Trees

Consider now a more general case of a p-ary tree of height h (Fig. 2). Here $n = p^{h-1}$ is the number of leaves. The check matrix H_h has the following form:

$$H_h = \begin{bmatrix} H_{h-1} & H_{h-1} & \cdots & & H_{h-1} & & \\ 11\cdots1 & \alpha\alpha\cdots\alpha & \cdots & \alpha^{p-1} & \alpha^{p-1} & \cdots & \alpha^{p-1} \end{bmatrix}, \tag{41}$$

where H_2 is given by (38) (with p substituted for n). The number of rows in this matrix $r = h$, whereas $r = 3h - 2$ for SD (cf. (25)). The location procedure is simple. Let $(S_1, S_2, \ldots, S_h)$ be the syndromes. Then, similar to the procedure in Section 5.1, the ratio S_h/S_1 reveals the faulty subtree of height $h-1$: if $S_h/S_1 = \alpha^i$, $i = 0, 1, \ldots, p-1$ we decide the fault is in the subtree whose root is the node i at the second level from the top (Fig. 2), otherwise the tree root is faulty. Similarly, if $S_{h-1}/S_1 = \alpha^i$, $i = 0, 1, \ldots, p-1$, it reduces the faulty part to a subtree of height $h-2$, etc.

However, there is a possibility of making a wrong decision at any step of the procedure, due to the fact that it may happen $S_{h-k}/S_1 = \alpha^i$, $i = 0, 1, \ldots, p-1$; $k = 0, 1, \ldots, h-2$, in spite of the fact that the fault is at level $k+1$ from the top. Using the expression (40) and omitting terms of higher order in q^{-1}, we obtain the fraction of misdiagnosis cases:

$$\begin{aligned} \omega_e &= pq^{-1} + (1-pq^{-1})pq^{-1} + (1-pq^{-1})^2 pq^{-1} + \cdots + (1-pq^{-1})^{h-2} pq^{-1} \\ &= 1 - (1-pq^{-1})^{h-1} \end{aligned} \qquad (42)$$

For large q, $\omega_e \approx (h-1)pq^{-1}$, which is, again, very small for practical values of the parameters.

5.3 Faults in Disconnected Processors

Consider now the situation when we have a set of n processors disconnected in the testing mode. Then a fault in a single processor manifests itself as a single error. Our goal is to locate faulty processors, i.e. to locate errors up to multiplicity l. Hence the set of errors is a ball B_l of radius l in V_q^n centered at the origin.

Obviously, the minimum number of check digits that would allow us to locate at least some errors is $r = l+1$. We choose matrix H to be the check matrix of a q-ary Reed-Solomon code:

$$H = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ 1 & \alpha^2 & \alpha^4 & \cdots & \alpha^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^l & \alpha^{2l} & \cdots & \alpha^{l(n-1)} \end{bmatrix}. \qquad (43)$$

The remarkable property of this matrix is that any $(l+1)$ its columns are linearly independent. Since Reed-Solomon is an MDS code, it attains the Singleton bound, and its distance $d = r+1 = l+2$.

It should be pointed out the difference between our problem of error location and the more usual problem of correct decoding [7], [21]. In spite of the fact that, in the situation considered, any code that locates all errors up to a given multiplicity corrects all such errors as well, the problems are quite different.

In decoding the goal is to find a coset leader of the minimum weight, thereby minimizing the probability of error. In our problem we do not consider probabilities; we are interested to find out how many errors from B_l can be uniquely identified by their syndromes, in other words, which cosets intersects with B_l in exactly one point.

Consider the set $\{Y_i\}$, $i = 1, 2, \ldots, \binom{n}{l}$, of all possible l-tuples from n. Then $V_i = \{E_{ik} | \mathrm{supp}(E_{ik}) \subseteq Y_i\}$ is an l-dimensional coordinate subspace in V_q^n, $|V_i| = q^l$. Obviously, $\bigcup_i V_i = B_l$ It is easy to see that the mapping $H : V_q^n \to V_q^{l+1}$ maps any V_i to V_q^{l+1} injectively. Indeed, the opposite would mean a linear dependence of l columns of H. Also we observe that

$$\sum_i |V_i| = \binom{n}{l} q^l \approx |B_l| = \sum_{m=0}^{l} \binom{n}{m} (q-1)^m. \tag{44}$$

The ratio of these numbers differ from 1 in terms of order $nl^{-1}q^{-1}$. Henceforth we neglect these terms assuming that $l \ll n \ll q$.

Denote the image $H(V_i) = W_i$. All W_i are l-dimensional subspaces in the $(l+1)$-dimensional space of syndromes $H(V_q^n) = \mathrm{W}$. Let us estimate how many members of W_i do not belong to any other subspace $W_j = H(V_j)$. Note that any two subspaces W_i, W_j intersect over a subspace of dimension $(l-1)$. The minimum dimension of intersection of three subspace is $(l-2)$, etc. (The probability that randomly chosen t subspaces intersect over a subspace of dimension higher than $l - t + 1$ is of order $q^{-(l-t+2)}$, $t \leq l + 1$). Alternatively subtracting and adding terms corresponding to intersections of two, three, etc. subspaces we obtain the following estimate for the fraction of errors in any W_i that can be located:

$$\begin{aligned} \omega_c &\approx q^{-l}\left[q^l - \left[\binom{n}{l} - 1\right] q^{l-1} + \binom{\binom{n}{l} - 1}{2} q^{l-2} - \cdots\right] \\ &= \sum_{t=0}^{l+1} (-1)^t \binom{\binom{n}{l} - 1}{t} q^{-t}, \end{aligned} \tag{45}$$

where $\omega_e = 1 - \omega_c$.

We are interested in the case when ω_c is close enough to 1. It requires that $\binom{n}{l} q^{-1} < 1$. Then the terms in (45) decrease monotonically and we can extend the summation up to $\binom{n}{l}$ terms. Moreover, since the largest intersection term (the second) is negative, one can believe that the expression will give us a lower bound for the fraction of localizable errors. Therefore,

$$\omega_c \geq \sum_{t=0}^{\binom{n}{l}-1} (-1)^t \binom{\binom{n}{l} - 1}{t} q^{-t} = \left(1 - q^{-1}\right)^{\binom{n}{l}-1} \approx e^{-\frac{\binom{n}{l}}{q}} \tag{46}$$

For example, if $l = 5$, $n = 100$, $q = 2^{32}$, then $\omega_c = 0.979$. Thus, by allowing a

Table 2: Minimal Numbers of Check Symbols for Weak Diagnosis of Multiprocessor Interconnection Networks.

Graph	Parameters N, n	Check Symbols r	Fractions of misdiagnosed errors ω_e
p-ary star	$p+1, p$	2	$(nq-1)q^{-2}$
p-ary full tree of height h	$\frac{p^h-1}{p-1}, p^{h-1}$	h	$1-(1-pq^{-1})^{h-1}$
n disconnected processors with up to l faults	n, n	$l+1$	$\leq 1-(1-q^{-1})^{\binom{n}{l}-1}$

small fraction of errors not to be located we reduce substantially the redundancy from $r = 2l + 1$ to $r = l + 1$.

In Table 2 we summarize our results on linear error locating codes for different meshes (stars, trees and disconnected) with weak diagnostic.

6 Conclusions

In this paper we have presented bounds on numbers of check symbols required for codes detecting and locating arbitrary set of errors. These codes can be used for identification of faulty processing elements in multiprocessor systems or array processors. We presented several nearly optimal error detecting and locating codes for tree and FFT errors. Hardware implementation based on the proposed codes results in considerable savings of redundant overhead. The concepts of strong and weak diagnostics are introduced and estimates of probabilities of correct fault location for weak diagnostics are presented.

Acknowledgment

The authors would like to thank Professor Tatyana D. Roziner of Boston University, Boston MA, for valuable discussions on the results presented in this paper.

References

[1] P. H. Bardell, W. H. McAnney, and J. Savir. *Built-in Self Test for VLSI: Pseudorandom Techniques.* Wiley Interscience, New York, NY, 1987.

[2] K. E. Batcher. Design of a Massively Parallel Processor. *IEEE Transaction on Computers*, C-29:836–840, Sept. 1980.

[3] J. Bently and H. T. Kung. A Tree Machine for Searching Problems. In *International Conference on Parallel Processing*, pages 257–266, 1979.

[4] C. Berge. *Graphs and Hypergraphs.* North-Holland, New York, NY, 1973.

[5] D. F. Elliott and K. R. Rao. *Fast Transforms: Algorithms, Analyses, Applications.* Academic Press, New York, NY, 1982.

[6] T. Y. Feng. A Survey of Interconnection Networks. *IEEE Computer*, 14:960–965, 1981.

[7] C. R. P. Hartmann. Decoding Beyond the BCH Bound. *IEEE Transaction on Information Theory*, pages 441–444, May 72.

[8] J. P. Hayes et al. A Microprocessor-Based Hypercube Supercomputer. *IEEE Micro*, 6:6–17, Oct. 1986.

[9] W. D. Hillis. *The Connection Machine.* MIT Press, Cambridge, MA, 1985.

[10] K. Hwang and F. A. Briggs. *Computer Architecture and Parallel Processing.* Academic Press, New York, NY, 1982.

[11] M. G. Karpovsky. Weight Distributions of Translates, Covering Radius, and Perfect Codes Correcting Errors of Given Weights. *IEEE Transaction on Information Theory*, IT-27:462–472, July 1981.

[12] M. G. Karpovsky and S. M. Chaudhry. Built-in Self Diagnostic by Space-Time Compression of Test Responses. In *IEEE VLSI Test Symposium*, pages 149–154, 1992.

[13] M. G. Karpovsky, L. B. Levitin, and F. S. Vainstein. Identification of Faulty Processing Elements by Space-Time Compression of Test Responses. In *International Test Conference*, pages 638–647, 1990.

[14] M. G. Karpovsky and V. D. Milman. On Subspaces Contained In Subsets Of Finite Homogeneous Space. *Discrete Mathematics*, 22:273–280, 1978.

[15] S. Y. Kung. *VLSI Array Processors.* Prentice-Hall, Englewood Cliffs, NJ, 1988.

[16] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes.* North-Holland, New York, NY, 1977.

[17] B. Masnick and J. Wolf. On Linear Unequal Error Protection Codes. *IEEE Transaction on Information Theory*, IT-3:600–607, Oct. 1967.

[18] E. J. McCluskey. Built-in Self Test Techniques. *IEEE Design and Test of Computers*, pages 21–28, Apr. 1985.

[19] F. P. Preparata and J. Vuillemin. The Cube-Connected Cycles: A Versatile Network for Parallel Computation. *Communication of the ACM*, 24:568–572, May 1981.

[20] S. R. Reddy, K. K. Saluja, and M. G. Karpovsky. A Data Compression Technique for Test Responses. *IEEE Transaction on Computers*, C-38:1151–1156, Sept. 1988.

[21] U. K. Sorger. A New Reed-Solomon Code Decoding Algorithm Based on Newton's Interpolation. *IEEE Transaction on Information Theory*, IT-39:358–365, Mar. 1993.