D. C. Bossen

b-Adjacent Error Correction

Abstract: A high-speed method is derived for single-symbol error correcting Reed-Solomon and Hamming type codes. A matrix description is used for implementation of the codes, in which single-error correction in the Galois field 2^b corresponds to correcting a block of b bits in a binary field. The resulting codes correct not only single-bit errors but also single clusters of b-adjacent-bit errors.

Introduction

In the application of error correcting codes to digital computer systems, there are a number of situations for which an error correcting code capable of correcting clusters of adjacent bits in error is uniquely suited. An example would be the error due to a failure in the addressing circuitry of a memory system that is packaged in a b-bits-per-card basis, where the number of bits in a memory word is equal to some kb. If a failure occurs, the resultant information readout from memory is likely to have a block of b bits in error. In this kind of application, it may be desirable to have an error correcting code capable of correcting all single-bit errors as well as all single clusters of b-adjacent-bit errors.

It is well known that many of the results from the theory or error correcting codes[1] apply to information that is coded from any finite field. In particular, we are interested in codes with symbols from the Galois field of 2^b elements, i.e., GF (2^b). The reason for this is twofold. In the first case, single-error correction in the field GF (2^b) is equivalent to correcting a block of b bits in the binary field. Secondly, the codes with symbols from GF (2^b) have generally low redundancy. In many cases, the theoretical bounds are achieved since each additional check symbol increases the distance of the code by 1.

The central problem addressed by this paper is the high speed and practical implementation of error correcting systems capable of correcting clusters of errors. There are a number of known classes of error correcting codes that have this property. Among these are the Reed-Solomon codes[2] and the redundant residue polynomial codes.[3] In the applications for which the present work is intended, the speed of decoding is critical as, for example, in a high-speed memory or data path. The hardware approaches to the decoding problem proposed by Peterson[1] are cyclic in nature and require a relatively long processing time. The Oldham-Chien-Tang[4] approach to error correction using Reed-Solomon codes is essentially a software technique. The high-speed parallel single-error correction method developed in this paper, being a hardware technique, has the time for correction measured in terms of a few logic levels rather than in terms of a few processor cycles.

The derivation of the single-symbol error correcting decoding method presented in this paper does not depend on the cyclic property of the codes involved, and, in fact, the method is clearly applicable to noncyclic codes. The essential and well-known algebraic fact is that GF (2^b) is always an extension field of GF (2). This is used to describe the codes in a matrix form such as that proposed by Cocke,[5] who used this representation to show how any encoding and decoding computations could be performed in the prime field rather than in the extension field. From this matrix description, we show how to construct the high speed implementation for the single-error-correcting Reed-Solomon and Hamming codes with symbols from GF (2^b).

Hamming-type codes over GF (2^b)

It is well known that a Hamming single-error correcting code can be constructed with symbols from any finite field. If F is such a field, then the parity-check matrix $\mathbf{H}$ for a single-error correcting code with elements from F is constructed as follows: Choose as columns of $\mathbf{H}$ all

The author is located at the IBM Systems Development Division Laboratory, Poughkeepsie, New York.

the nonzero r-tuples of elements from F such that no two columns of $\mathbf{H}$ are linear multiples of each other in the field F. This can be accomplished either by deleting all but one of the columns that are linear multiples of each other or by keeping separate account of the linear multiples of every column as it is added to $\mathbf{H}$ so that new columns are not chosen from this set. This is easy to do once the multiplication and addition rules in F are known. Since no linear combination of $d - 1 = 2$ or fewer columns of $\mathbf{H}$ is equal to 0, the code has $d = 3$ and is capable of correcting any single error. If k such columns are found, an $(n, k) = (k + r, k)$ code results, where n is the number of symbols per word, k is the number of information bits per word and r is the number of check symbols per word.

In particular, if we use the field GF (2), then each symbol from GF (2^b) is equivalent to a binary b-tuple, and hence all b-adjacent errors occurring in the blocks corresponding to the elements of GF (2^b) can be corrected as long as only one such block of b errors occurs.

Example 1: A code with symbols from GF (2^2). In this example we use GF (2^2) = GF (2)[X] mod $p(X)$, where $p(X)$ is the irreducible polynomial $X^2 + X + 1$. In this case, GF (2^2) consists of the elements 0, 1, α, α^2, which correspond to the residue classes modulo $p(X)$ of 0, 1, X, $X + 1$. In vector form over GF (2) these elements or residue classes correspond to the vectors

$$\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}1\\1\end{bmatrix}.$$

The addition and multiplication rules in GF (2^2) are determined by $p(X)$, and are seen to be

+	0	1	α	α^2
0	0	1	α	α^2
1	1	0	α^2	α
α	α	α^2	0	1
α^2	α^2	α	1	0

×	0	1	α	α^2
0	0	0	0	0
1	0	1	α	α^2
α	0	α	α^2	1
α^2	0	α^2	1	α

These rules are used to test for linear dependence relations among r-tuples from GF (2^2). With these relations, it is easy to verify that the $\mathbf{H}$ matrix for a single-error-correcting code over GF (2^2) is as shown in Fig. 1. This code could be used for single-card correction with a 2 bit-per-card memory consisting of 64 data bits and 8 check bits.

Matrix description of error correcting codes over GF (2^b).

For the purpose of implementing the Hamming-type codes of the previous Section and the codes to be discussed in later Sections, it is necessary to obtain the $\mathbf{H}$ matrix in

Figure 1 Parity check matrix, H, for a single-error-correcting code from GF(2^2).

binary form rather than as symbols from GF (2^b). The essential fact necessary for accomplishing this is that the field GF (2^b) is a vector space of dimension b over the field GF (2). Addition of two elements in the field GF (2^b) corresponds to the bit-by-bit modulo 2 addition of their corresponding vector representations. Multiplication in GF (2^b) on the other hand can be thought of as defining a set of linear transformations on the corresponding vector space. Let β be the binary vector representation for an element β in GF (2^b). Then a linear transformation T_β is defined by

$$T_\beta(\gamma) = \beta\gamma \tag{1}$$

for all γ in GF (2^b), where the expression on the right-hand side of Eq. (1) is the vector representation of the element $\beta\gamma$ of GF (2^b). Each such linear transformation can be represented by a $b \times b$ matrix with elements from GF (2).

In particular, the identity element 1 of GF (2^b) is equivalent to the $b \times b$ identity matrix. Similarly, the 0 element is equivalent to the $b \times b$ 0-matrix.

Since GF (2^b) is equivalent to the residue class ring [GF (2)X] mod $p(X)$, where $p(X)$ is an irreducible polynomial of degree b over GF (2), we can consider the vector space corresponding to GF (2^b) to be spanned by the vectors

$$\mathbf{X}^{b-1}, \mathbf{X}^{b-2}, \cdots, \mathbf{X}, \quad 1$$

or

$$\begin{bmatrix}1\\0\\0\\\vdots\\0\end{bmatrix} \begin{bmatrix}0\\1\\0\\\vdots\\0\end{bmatrix}, \cdots, \begin{bmatrix}0\\0\\\vdots\\1\\0\end{bmatrix} \begin{bmatrix}0\\0\\\vdots\\0\\1\end{bmatrix}.$$

Then a matrix $\mathbf{T}_\beta$ corresponding to the linear transformation T_β is given by the concatenation of the columns

$$\mathbf{X}^{b-1}\beta, \mathbf{X}^{b-2}\beta, \cdots, \mathbf{X}\beta, \beta$$

so that

$$\mathbf{T}_\beta = [\mathbf{X}^{b-1}\beta, \mathbf{X}^{b-2}\beta, \cdots, \beta]. \tag{2}$$

With this definition, it is clear that multiplication in GF (2^b) of some β_2 by β_1 is equivalent to the ordinary

vector-by-a-matrix multiplication of the vector β_2 by the matrix $\mathbf{T}_{\beta_1}$.

Finally, the desired binary H matrix for a given code is constructed by replacing each β in GF (2^b) by the $b \times b$ matrix $\mathbf{T}_\beta$.

Example 2 (Continuation of Example 1): With the definition of GF (2^2) given in Example 1, the 2×2 matrices $\mathbf{T}_0$, $\mathbf{T}_1$, $\mathbf{T}_\alpha$, $\mathbf{T}_\alpha^2$ are as shown in Eq. (3).

$$\mathbf{T}_0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{T}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$
$$\mathbf{T}_\alpha = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{T}_\alpha^2 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}. \tag{3}$$

Substitution of these matrices into the matrix of Fig. 1 yields the binary H-matrix.

2-redundant b-adjacent codes

There is an interesting subclass of Hamming-type codes over GF (2^b) that have two check symbols and are capable of single-error correction in GF (2^b). These codes are called *2-redundant* for this reason, and they will always have a parity check matrix of the form

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1, \cdots, & 1 & 1 & 0 \\ 1 & \beta_1 & \beta_2, \cdots, \beta_{k-1} & 0 & 1 \end{bmatrix}. \tag{4}$$

If $\beta_1, \beta_2, \cdots, \beta_{k-1}, 1$ are all distinct elements of GF (2^b), it is clear that no two columns of H are linearly dependent so that the code has distance 3 and is, therefore, a single-error-correcting code. The maximum number of information symbols for such a code is equal to the number of nonzero elements in GF (2^b), which is $2^b - 1$. If the polynomial $p(X)$ defining GF (2^b) is chosen to be a primitive polynomial, then all powers of $\alpha = \{X\}$ are distinct so that every β in GF (2^b) is equal to some power of α. The resulting H-matrix can then have the form

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1, \cdots, & 1 & 1 & 0 \\ \alpha^0 & \alpha & \alpha^2, \cdots, \alpha^{2^b-2} & 0 & 1 \end{bmatrix}. \tag{5}$$

It will be shown in a later section that this form is particularly useful for implementation purposes.

Example 3: An (80, 64) code over GF (2^8). The proposed code is for a data length of 64 bits configured in eight 8-bit cards. Two additional "check cards" (16 bits) are required for error correction. Only one "check card" of 8 bits is required if it is desired only to detect a single card in error. A single card in error implies any error pattern whatever, as long as only a single card has an error. This could be total failure of the card or only some of the bits in error.

Let $\alpha, \alpha^2, \cdots, \alpha^7$ be the vector representation of distinct nonzero and nonidentity elements of GF (2^8). However, for simplicity of decoding, that is, for a minimum number of inputs to the EXCLUSIVE-OR gates in the decoder, it is very likely that there will be an optimal choice for the α^i. For simplicity let us choose

$$\begin{aligned} \alpha &= \{X\} \\ \alpha^2 &= \{X\}^2 \\ \alpha^3 &= \{X\}^3 \\ &\vdots \\ \alpha^7 &= \{X\}^7. \end{aligned} \tag{6}$$

If we use GF (2^8) = GF (2)[X] mod $(X^8 + X^4 + X^3 + X^2 + 1)$, which determines the multiplication and addition rules between elements of GF (2^8), then the matrices $\mathbf{T}_\alpha$, are given by (7).

$$\mathbf{T}_\alpha = \mathbf{T}_x = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{T}_\alpha^2 = \mathbf{T}_x^2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{T}_\alpha^3 = \mathbf{T}_x^3 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{T}_\alpha^4 = \mathbf{T}_x^4 = \begin{bmatrix} 1&0&0&0&1&0&0&0 \\ 1&1&0&0&0&1&0&0 \\ 1&1&1&0&0&0&1&0 \\ 0&1&1&1&0&0&0&1 \\ 1&0&1&1&0&0&0&0 \\ 0&1&0&1&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \end{bmatrix},$$

$$\mathbf{T}_\alpha^5 = \mathbf{T}_x^5 = \begin{bmatrix} 1&1&0&0&0&1&0&0 \\ 1&1&1&0&0&0&1&0 \\ 0&1&1&1&0&0&0&1 \\ 0&0&1&1&1&0&0&0 \\ 1&1&0&1&1&0&0&0 \\ 1&0&1&0&1&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 1&0&0&0&1&0&0&0 \end{bmatrix}, \qquad (7)$$

$$\mathbf{T}_\alpha^6 = \mathbf{T}_x^6 = \begin{bmatrix} 1&1&1&0&0&0&1&0 \\ 0&1&1&1&0&0&0&1 \\ 0&0&1&1&1&0&0&0 \\ 0&0&0&1&1&1&0&0 \\ 0&1&1&0&1&1&0&0 \\ 1&1&0&1&0&1&0&0 \\ 1&0&0&0&1&0&0&0 \\ 1&1&0&0&0&1&0&0 \end{bmatrix},$$

$$\mathbf{T}_\alpha^7 = \mathbf{T}_x^7 = \begin{bmatrix} 0&1&1&1&0&0&0&1 \\ 0&0&1&1&1&0&0&0 \\ 0&0&0&1&1&1&0&0 \\ 1&0&0&0&1&1&1&0 \\ 0&0&1&1&0&1&1&0 \\ 0&1&1&0&1&0&1&0 \\ 1&1&0&0&0&1&0&0 \\ 1&1&1&0&0&0&1&0 \end{bmatrix},$$

$$\mathbf{T}_1 = I_8, \qquad \mathbf{T}_0 = 0_8.$$

It is evident that the **H**-matrix of a single-error correcting code over GF (2^8) is, according to Eq. (5),

$$\mathbf{H} = \begin{bmatrix} 1&1&1&1&1&1&1&1&1&0 \\ 1&\alpha^1&\alpha^2&\alpha^3&\alpha^4&\alpha^5&\alpha^6&\alpha^7&0&1 \end{bmatrix}. \qquad (8)$$

Substituting the matrices $\mathbf{T}_{\alpha^i}$ for α^i in (8) where the $\mathbf{T}_{\alpha^i}$ are given in (7) results in the binary **H**-matrix of Fig. 2.

Reed-Solomon codes

The Reed-Solomon codes are a general class of cyclic multiple-error correcting codes over GF (2^b). In this Section, a matrix description of these codes is given along with a modified form for implementation.

In Ref. 1 it is shown that the generator polynomial for a distance d code with symbols from GF (2^b) is given by

$$g(X) = (X - \alpha)(X - \alpha^2), \cdots, (X - \alpha^{d-1}), \qquad (9)$$

where we will choose α to be a primitive element of GF (2^b). It has previously been shown that any element in GF (2^b) can be represented by a $b \times b$ binary matrix. The **H**-matrix of the code specified by Eq. (9) can then be written as

$$\mathbf{H} = \begin{bmatrix} \begin{pmatrix} \\ \\ \\ \end{pmatrix} & \begin{pmatrix} \\ \\ \\ \end{pmatrix} & \cdots & \begin{pmatrix} \\ \\ \\ \end{pmatrix} \\ r_{n-1}\ r_{n-2} & \cdots & r_{d-1}\ r_{d-2}, \cdots, r_1\ r_0 \end{bmatrix} = \begin{bmatrix} 1, \cdots, 0\ 0 \\ 0 \\ \vdots \\ 0, \cdots, 1\ 0 \\ 0, \cdots, 0\ 1 \end{bmatrix}, \qquad (10)$$

where $n = 2^b - 1$ and $\mathbf{1}$ is the $b \times b$ identity matrix and $\mathbf{r}_i$ is a column vector whose entries are $b \times b$ matrices that represent the coefficients of the remainder polynomial after dividing X^i by $g(X)$ in GF (2^b). That is, since each residue $\mathbf{r}_i$ has the form

$$\mathbf{r}_i = r_{i0}X^0 + r_{i1}X + r_{i2}X^2 + \cdots + r_{i(d-2)}X^{d-2}, \qquad (11)$$

where each r_{ij} is an element of GF (2^b), then a matrix representation of the column vector

$$\mathbf{r}_i = \begin{bmatrix} \mathbf{r}_{i(d-2)} \\ \vdots \\ r_{i1} \\ \mathbf{r}_{i0} \end{bmatrix} \qquad (12)$$

is a column vector of $b \times b$ matrices corresponding to the r_{ij}. An example from GF (2^3) should make these ideas clear.

Example 4: Reed-Solomon Code over GF (2^3). The addition and multiplication rules for GF (2^3) are determined by the polynomial

$$p(X) = X^3 + X^2 + 1. \qquad (13)$$

Figure 2 H-matrix for the (80, 64) 8-adjacent-error correcting code.

Table 1 Addition in GF (2^3).

+	1	T_α	$T_\alpha{}^2$	$T_\alpha{}^3$	$T_\alpha{}^4$	$T_\alpha{}^5$	$T_\alpha{}^6$
1	0	$T_\alpha{}^3$	$T_\alpha{}^6$	T_α	$T_\alpha{}^5$	$T_\alpha{}^4$	$T_\alpha{}^2$
T_α	$T_\alpha{}^3$	0	$T_\alpha{}^4$	1	$T_\alpha{}^2$	$T_\alpha{}^6$	$T_\alpha{}^5$
$T_\alpha{}^2$	$T_\alpha{}^6$	$T_\alpha{}^4$	0	$T_\alpha{}^5$	T_α	$T_\alpha{}^3$	1
$T_\alpha{}^3$	T_α	1	$T_\alpha{}^5$	0	$T_\alpha{}^6$	$T_\alpha{}^2$	$T_\alpha{}^4$
$T_\alpha{}^4$	$T_\alpha{}^5$	$T_\alpha{}^2$	T_α	$T_\alpha{}^6$	0	1	$T_\alpha{}^3$
$T_\alpha{}^5$	$T_\alpha{}^4$	$T_\alpha{}^6$	$T_\alpha{}^3$	$T_\alpha{}^2$	1	0	T_α
$T_\alpha{}^6$	$T_\alpha{}^2$	$T_\alpha{}^5$	1	$T_\alpha{}^4$	$T_\alpha{}^3$	T_α	0

If we make $\alpha = \{X\}$ be a primitive element of GF (2^3) by picking $p(X)$ to be primitive, then every element of GF (2^3) is equal to a power of α, so that the multiplication rules are trivial. The addition rules are shown in Table 1.

According to (9) we chose as the generator polynomial of a distance 4 code

$$\begin{aligned} g(X) &= (X - \mathbf{T})(X - \mathbf{T}^2)(X - \mathbf{T}^3) \\ &= X^3 + X^2(\mathbf{T} + \mathbf{T}^2 + \mathbf{T}^3) \\ &\quad + X(\mathbf{T}^5 + \mathbf{T}^4 + \mathbf{T}^3) + \mathbf{T}^6. \end{aligned} \tag{14}$$

By applying the addition rules of Table 1, this reduces to

$$g(X) = X^3 + X^2(\mathbf{T}^6) + X(\mathbf{T}) + \mathbf{T}^6, \tag{15}$$

where the $\mathbf{T}^i$ represent elements of GF (2^b) and $\alpha = \{X\}$ is primitive. Choosing $\alpha = \{X\}$ in GF (2^b) results in T_α being the companion matrix of the polynomial $p(X) = X^3 + X^2 + 1$.

Then $n = 2^3 - 1 = 7$, and we have

$$\mathbf{H} = \begin{array}{c} \begin{matrix} r_6 & r_5 & r_4 & r_3 & r_2 & r_1 & r_0 \end{matrix} \\ \begin{bmatrix} \mathbf{T} & \mathbf{T}^3 & \mathbf{T}^6 & \mathbf{T}^6 & 1 & 0 & 0 \\ 1 & \mathbf{T}^4 & \mathbf{T}^2 & \mathbf{T} & 0 & 1 & 0 \\ \mathbf{T}^2 & \mathbf{T}^5 & \mathbf{T}^5 & \mathbf{T}^6 & 0 & 0 & 1 \end{bmatrix} \end{array}. \tag{16}$$

Finally, the binary form of (16) is obtained by replacing the symbol 1 by the $b \times b$ identity matrix, and by replacing $\mathbf{T}$ by the $b \times b$ companion matrix of $p(X) = X^3 + X^2 + 1$.

For implementation purposes it may be desirable to have the resultant parity-check matrix of a form in which all identity matrices are in the top row, similar to the matrix of (5). This is easily accomplished using the following lemma.

Lemma: Each column of $\mathbf{H}$ can be normalized so that the top entry is 1, the identity of GF (2^b).

Proof: Consider a set of $d - 1$ columns of $\mathbf{H}$, where for example, $d = 4$,

$$\begin{bmatrix} \mathbf{T}^{i1} \\ \mathbf{T}^{i2} \\ \mathbf{T}^{i3} \end{bmatrix}, \quad \begin{bmatrix} \mathbf{T}^{j1} \\ \mathbf{T}^{j2} \\ \mathbf{T}^{j3} \end{bmatrix}, \quad \begin{bmatrix} \mathbf{T}^{k1} \\ \mathbf{T}^{k2} \\ \mathbf{T}^{k3} \end{bmatrix}. \tag{17}$$

These columns are linearly independent over GF (2^b). It is also claimed that the columns

$$\begin{bmatrix} 1 \\ \mathbf{T}^{i2-i1} \\ \mathbf{T}^{i3-i1} \end{bmatrix}, \quad \begin{bmatrix} 1 \\ \mathbf{T}^{j2-j1} \\ \mathbf{T}^{j3-j1} \end{bmatrix}, \quad \begin{bmatrix} 1 \\ \mathbf{T}^{k2-k1} \\ \mathbf{T}^{k3-k1} \end{bmatrix} \tag{18}$$

are linearly independent. If not, then there exists some $\mathbf{T}^{P_i}$, $\mathbf{T}^{P_j}$, $\mathbf{T}^{P_k} \neq 0$ such that

$$\mathbf{T}^{P_i}\begin{bmatrix} 1 \\ \mathbf{T}^{i2-i1} \\ \mathbf{T}^{i3-i1} \end{bmatrix} + \mathbf{T}^{P_j}\begin{bmatrix} 1 \\ \mathbf{T}^{j2-j1} \\ \mathbf{T}^{j3-j1} \end{bmatrix} + \mathbf{T}^{P_k}\begin{bmatrix} 1 \\ \mathbf{T}^{k2-k1} \\ \mathbf{T}^{k3-k1} \end{bmatrix} = 0. \tag{19}$$

But then

$$\mathbf{T}^{P_{i+i_1}}\begin{bmatrix} \mathbf{T}^{i1} \\ \mathbf{T}^{i2} \\ \mathbf{T}^{i3} \end{bmatrix} + \mathbf{T}^{P_{j+j_1}}\begin{bmatrix} \mathbf{T}^{j1} \\ \mathbf{T}^{j2} \\ \mathbf{T}^{j3} \end{bmatrix} + \mathbf{T}^{P_{k+k_1}}\begin{bmatrix} \mathbf{T}^{k1} \\ \mathbf{T}^{k2} \\ \mathbf{T}^{k3} \end{bmatrix} = 0 \tag{20}$$

contrary to the linear independence of (18)! Q.E.D.

Using now the fact that each column can be normalized (by multiplication of each element by the inverse of the top element) without affecting the linear independence of combinations of the columns, the parity-check matrix (16) of Example 4 becomes

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ \mathbf{T}^6 & \mathbf{T} & \mathbf{T}^3 & \mathbf{T}^2 & 0 & 1 & 0 \\ \mathbf{T} & \mathbf{T}^2 & \mathbf{T}^6 & 1 & 0 & 0 & 1 \end{bmatrix}. \tag{21}$$

These procedures are quite general and can be used to generate useful descriptions of codes for any specified error-correcting capability and any specified length less than or equal to the natural length of the code.

Implementation of single-error correction in GF (2^b)

• *Decoding of Hamming-type codes in GF (2^b)*

A possible method of implementing single-error correction with the Hamming-type codes is to use a set of $(2^b - 1)N$ AND gates to recognize the possible syndrome patterns. Here N is the length of the code word in symbols from GF (2^b). In the case of the (72, 64) GF (2^2) code of Examples 1 and 2 this would require 216 8-way AND gates.

There is an alternate method of performing the error correction that reveals a clear-cut cost vs speed trade-off. The syndrome is computed by using an EXCLUSIVE-OR tree in the conventional manner. The syndrome is equal to

$$\mathbf{S} = (S_1, S_2, \cdots, S_r). \tag{22}$$

If the ith column of $\mathbf{H}$ is equal to

$$\begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_r \end{bmatrix}, \tag{23}$$

then an error e_i in symbol i yields a syndrome that is equal to

$$\beta_1 e_i, \beta_2 e_i, \cdots, \beta_r e_i. \tag{24}$$

Since e_i is an element of GF (2^b), it is equal to some power of $\alpha = \{X\}$. Then there is some power j such that $\alpha^j e_i = 1$. Therefore, if the $(S_1, S_2, \cdots, S_r)$ are loaded simultaneously into a set of r linear feedback shift registers connected according to the companion matrix of $p(X)$ which defines GF (2^b), then after j shifts, the contents of the set of shift registers will be $(\beta_1, \beta_2, \cdots, \beta_r)$ which can be recognized by a single AND gate. Since there are N columns, the total number of AND gates is equal to N. This gives the error location. Since every column of $\mathbf{H}$ contains at least one identity element, then the original syndrome contains the error magnitude in one of the positions $S_1, S_2, \cdots, S_r$. This is gated in a bit-by-bit EXCLUSIVE-OR to the error location to produce the correct information. A block diagram of this implementation is shown in Fig. 3.

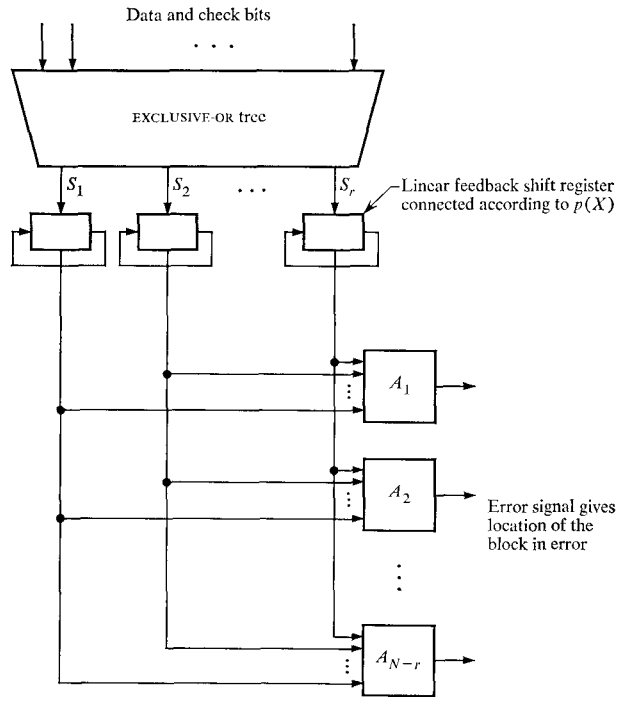

Figure 3 Block diagram of single-error correction scheme.

• *Decoding of 2-redundant codes*

An important characteristic of the 2-redundant codes discussed earlier lies in the method by which they may be decoded. The decoding procedure is first discussed in general and then the decoder for the (80, 64) code of Example 3 is presented.

Suppose that an error of b or fewer bits occurs in block i of the data bits. This error pattern corresponds to some $e_i \in$ GF (2^8). The syndrome that corresponds to this error has the value

$$\begin{bmatrix} S_1 \\ S_2 \end{bmatrix} = \begin{bmatrix} e_i \\ \alpha^i e_i \end{bmatrix} = \text{syndrome}, \tag{25}$$

where S_1 and S_2 are binary column vectors of b components. This can be seen from (5). If the error occurs in either of the check blocks, then the syndrome has the value

$$\begin{bmatrix} S_1 \\ S_2 \end{bmatrix} = \begin{bmatrix} e_i \\ 0 \end{bmatrix} \tag{26}$$

or

$$\begin{bmatrix} S_1 \\ S_2 \end{bmatrix} = \begin{bmatrix} 0 \\ e_i \end{bmatrix}$$

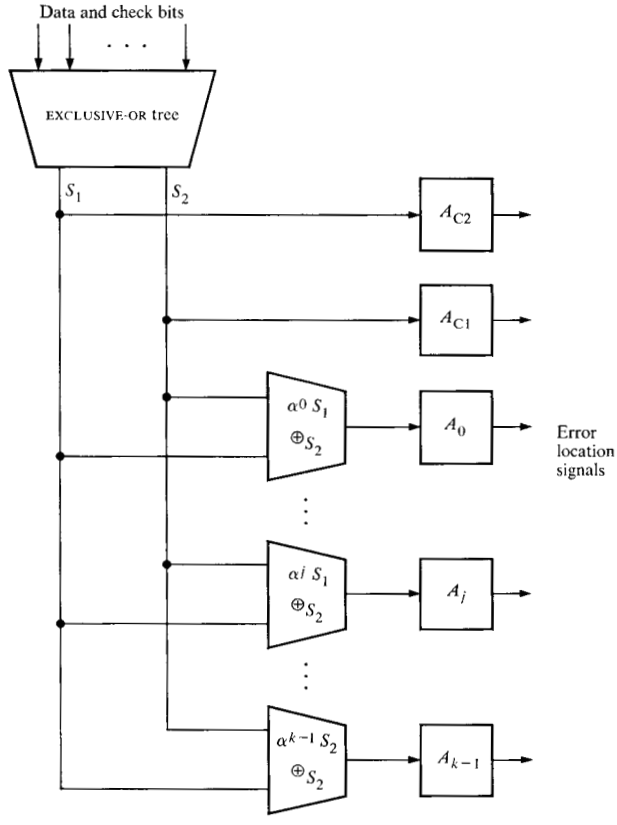

Figure 4 Basic error-correcting scheme for the 2-redundant codes.

depending on whether the error is in the first or second check block. The cases given in (26) are easily detected by AND gates checking either for S_1 = all 0's or S_2 = all 0's. This is true since the case given by (25) can never have e_1 or e_2 equal to 0, because in GF (2^b) $e \neq 0 \Rightarrow \alpha^i e \neq 0$, $\forall\ \alpha^i \in$ GF (2^b): $\alpha \neq 0$.

For a syndrome as in (25), it can be shown that the error is in block i if and only if

$$\alpha^i S_1 = S_2\ , \tag{27}$$

in which case the error of value $e_i = S_1$ can be added mod 2 to block i for error correction. Testing for the conditions specified in (26) and (27) forms the basis for the error correction. Eq. (27) can be rewritten as

$$\alpha^i S_1 + S_2 = 0\ , \tag{28}$$

where + stands for bit-by-bit EXCLUSIVE-OR.

A set of i EXCLUSIVE-OR circuits, $i = 1, 2, \cdots, 2^b - 1$, can be built to test for condition (28) using a circuit like that shown in Fig. 4.

It is easily seen from Fig. 4 that this method of decoding requires the following circuitry in addition to that needed for generation of the syndrome bits. K is the number of data blocks.

$K + 2$ AND gates of b inputs
Kb EXCLUSIVE-OR gates with an average of $b/2 + 1$ inputs each.

As a particular example, we have for the (80, 64) 8-adjacent error correcting code the following:

10 AND gates of 8 inputs
64 EXCLUSIVE-OR gates with average of 5 inputs.

Straightforward AND-gate decoding of the syndrome would require 2040 AND gates of 16 inputs.

Conclusions

Error correction systems that have the characteristic of high-speed parallel implementation are required in computer applications. Codes that are capable of correcting blocks of errors have a good potential for use due to their generally low redundancy. However, their decoding speed must be competitive with other high-speed error correction techniques such as Hamming single-error correction. To this end, a high-speed method for implementing single-symbol error correction with the Reed-Solomon and Hamming type codes with symbols from GF (2^b) has been derived, using a matrix description of these codes. It is felt that the matrix description of codes over GF (2^b) can be used to produce many other useful results.

References

1. W. W. Peterson, *Error Correcting Codes,* M.I.T. Press, Cambridge, Mass., 1961.
2. I. S. Reed and G. Solomon, "Polynomial Codes Over Certain Finite Fields," *Journal of S.I.A.M.* **8,** 300 (1960).
3. D. C. Bossen and S. S. Yau, "Redundant Residue Polynomial Codes," *Inf. and Control* **13,** 597 (1968).
4. I. B. Oldham, R. T. Chien and D. T. Tang, "Error Detection and Correction in a Photo-digital Storage System," *IBM J. Res. Develop.* **12,** 422 (1968).
5. J. Cocke, "Lossless Symbol Coding with Non-Primes," *IEEE Trans. on Inf. Theory* **IT-5,** 33 (1959).

Received October 21, 1969