N84-29542

Reed Solomon Codes for Error Control in Byte Organized Computer Memory Systems

Shu Lin

Dept. of Electrical Engineering
University of Hawaii
Honolulu, HI 96822

Daniel J. Costello, Jr.

Dept. of Comp. & Electrical Engineering
Illinois Institute of Technology
Chicago, IL 60616

Final Report

Phase II

NASA Grant NAG 2-202

July, 1984

This report contains two parts:

Part I - Tutorial material on Reed Solomon Codes

Part II - A manuscript entitled "Reed Solomon Codes for Error Control in Byte Organized Computer Memory Systems"

PART I

REED-SOLOMON CODES

For any $m \leq 3$ and any $t < 2^m$, there exists a t-error-correcting RS code with code symbols from $GF(2^m)$. The code has the following parameters:

$$n = 2^m - 1$$

$$n - k = 2t$$

$$d_{min} = 2t + 1$$

The code is capable of correcting any t or fewer symbol errors over a span of n symbols.

Let α be a primitive element in $GF(2^m)$. The generator polynomial is

$$\bar{g}(X) = (X + \alpha)(X + \alpha^2)\cdots(X + \alpha^{2t})$$

$$= g_0 + g_1 X + \cdots + g_{2t-1} X^{2t-1} + X^{2t}$$

where $g_i \in GF(2^m)$.

**A polynomial of degree n - 1 or less with coefficients from $GF(2^m)$ is a code polynomial if and only if it is a multiple of $\bar{g}(X)$.

**Every code polynomial has α, $\alpha^2\cdots\alpha^{2t}$ as roots.

ENCODING

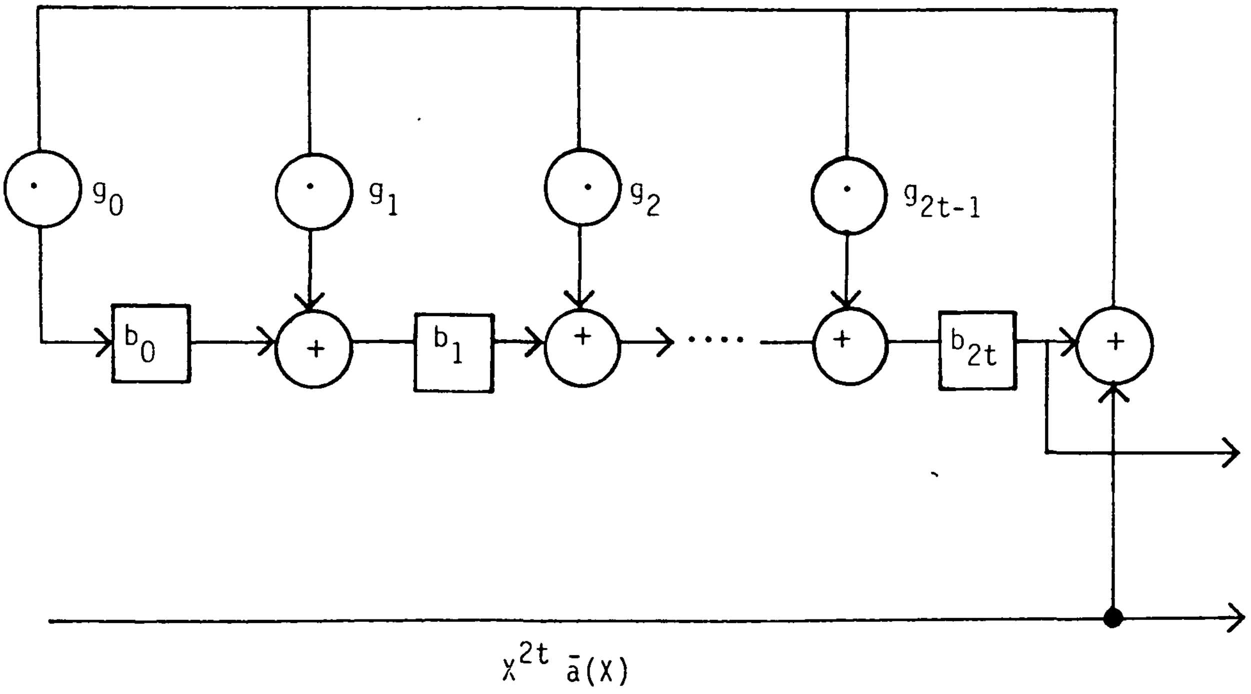
*Let $\bar{a}(X) = a_0 + a_1 X + \cdots + a_{k-1} X^{k-1}$ be the message to be encoded.

*Dividing $X^{2t}\ \bar{a}(X)$ by $\bar{g}(X)$

$$X^{2t}\ \bar{a}(X) = \bar{c}(X)\ \bar{g}(X) + \bar{b}(X)$$

*The code polynomial for $\bar{a}(X)$ is

$$\bar{b}(X) + X^{2t}\ \bar{a}(X) = \bar{c}(X)\ \bar{g}(X).$$



DECODING

*Received polynomial

$$\bar{r}(X) = r_0 + r_1 X + \cdots + r_{n-1} X^{n-1}$$

*Then

$$\bar{r}(X) = \bar{v}(X) + \bar{e}(X)$$

*Syndrome

$$\bar{S} = (S_1, S_2, \cdots, S_{2t})$$

where

$$S_i = \bar{r}(\alpha^i).$$

*If $\bar{S} = \bar{0}$, $\bar{r}(X)$ is a code polynomial. If $\bar{S} \neq \bar{0}$, $\bar{r}(X)$ is not a code polynomial and the presence of errors is detected.

SYNDROME COMPUTATION

*Dividing $\bar{r}(X)$ by $(x + \alpha^i)$,

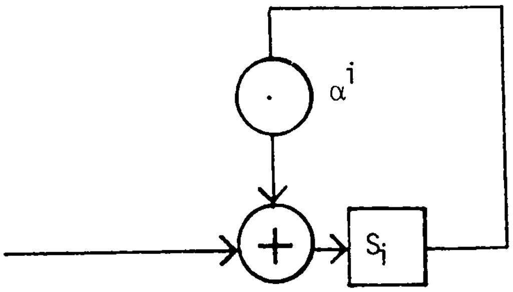
$$\bar{r}(X) = a_i(X)(X + \alpha^i) + b_i$$

where $b_i \varepsilon GF(2^m)$. Then

$$S_i = \bar{r}(\alpha^i) = b_i$$

for $i = 1, 2, \cdots, 2t$

*Circuit



Note that $\bar{r}(X) = \bar{v}(X) + \bar{e}(X)$ and $\bar{v}(X)$ has $\alpha, \alpha^2, \cdots \alpha^{2t}$ as roots. Then

$$S_i = \bar{r}(\alpha^i) = \bar{v}(\alpha^i) + \bar{e}(\alpha^i)$$

$$S_i = \bar{e}(\alpha^i) \qquad (1)$$

for $i = 1, 2, \cdots, 2t$.

Suppose that the error pattern contains ν symbol errors,

$$\bar{e}(X) = e_{j_1} X^{j_1} + e_{j_2} X^{j_2} + \cdots + e_{j_\nu} X^{j_\nu} \qquad (2)$$

where $0 \leq j_1 < j_2 < \cdots < e_{j\nu} <$ and $e_{j_\ell} \varepsilon GF(2^m)$.

From (1) and (2), we have

$$\begin{aligned} S_1 &= e_{j_1}\alpha^{j_1} + e_{j_2}\alpha^{j_2} + \cdots + e_{j_\nu}\alpha^{j_\nu} \\ S_2 &= e_{j_1}\alpha^{2j_1} + e_{j_2}\alpha^{2j_2} + \cdots + e_{j_\nu}\alpha^{2j_\nu} \\ &\vdots \\ S_{2t} &= e_{j_1}\alpha^{2tj_1} + e_{j_2}\alpha^{2tj_2} + \cdots e_{j_\nu}\alpha^{2tj_\nu} \end{aligned} \qquad (3)$$

Note that ν, j_ℓ, e_{j_ℓ} are all unknown.

*Let

$$\beta_\ell = \alpha^{j_\ell}$$

for $\ell = 1, 2, \cdots, \nu$.

*β_1, $\beta_2, \cdots$, β_ℓ are called error location numbers.

Now,

$$\begin{aligned} S_1 &= e_{j_1} \beta_1 + e_{j_2} \beta_2 + \cdots + e_{j_\nu} \beta_\ell^{2t} \\ S_2 &= e_{j_1} \beta_1^2 + e_{j_2} \beta_2^2 + \cdots e_{j_\nu} \beta_\nu^2 \\ &\vdots \\ S_{2t} &= e_{j_1} \beta_1^{2t} + e_{j_2} \beta_2^{2t} + \cdots + e_{j_\nu} \beta_\nu^{2t} \end{aligned} \tag{4}$$

Error-Location Polynomial

$$\begin{aligned} \sigma(X) &= (1 + \beta_1 X)(1 + \beta_2 X)\cdots(1 + \beta_\nu X) \\ &= 1 + \sigma_1 X + \sigma_2 X^2 + \cdots + \sigma_\nu X^\nu \end{aligned}$$

Note that

$\sigma(X)$ has $\beta_1^{-1}, \beta_2^{-1}, \cdots, \beta_\nu^{-1}$ as roots.

$$\begin{aligned}
\sigma_1 &= \beta_1 + \beta_2 + \cdots + \beta_\nu \\
\sigma_2 &= \beta_1 \beta_2 + \beta_1 \beta_3 + \cdots + \beta_{\nu-1} \beta_\nu \\
&\vdots \\
\sigma_\nu &= \beta_1 \beta_2 \cdots \beta_\nu
\end{aligned} \tag{5}$$

$$\begin{aligned}
S_{\nu+1} + \sigma_1 S_\nu + \sigma_2 S_{\nu-1} + \cdots + \sigma_\nu S_1 &= 0 \\
S_{\nu+2} + \sigma_1 S_{\nu+1} + \sigma_2 S_\nu + \cdots + \sigma_\nu S_2 &= 0 \\
&\vdots \\
S_{2t} + \sigma_1 S_{2t-1} + \sigma_2 S_{2t-2} + \cdots + \sigma_\nu S_{2t-\nu} &= 0
\end{aligned} \tag{6}$$

ITERATIVE METHOD

$\sigma(X)$ can be found iteratively in 2t steps. Let

$$\sigma^{(\mu)}(X) = \sigma_0^{(\mu)} + \sigma_1^{(\mu)} X + \cdots + \sigma_{\ell_\mu}^{(\mu)} X^{\ell_\mu}$$

be a minimum degree polynomial whose coefficients staisfy the following $\mu - \ell_\mu$ identities:

$$\begin{gathered}
S_{\ell_\mu+1} + \sigma_1^{(\mu)} S_{\ell_\mu} + \cdots + \sigma_{\ell_\mu}^{(\mu)} S_1 = 0 \\
S_{\ell_\mu+2} + \sigma_1^{(\mu)} S_{\ell_\mu+1} + \cdots + \sigma_{\ell_\mu}^{(\mu)} S_2 = 0 \\
\vdots \\
S_\mu + \sigma_1^{(\mu)} S_{\mu-1} + \cdots + \sigma_{\ell_\mu}^{(\mu)} S_{\mu-\ell_\mu} = 0
\end{gathered}$$

Then we compute

$$d_{\mu} = S_{\mu+1} + \sigma_{1}^{(\mu)} S_{\mu} + \cdots + \sigma_{\ell_{\mu}}^{(\mu)} S_{\mu+1-\ell_{\mu}}$$

If $d_{\mu} = 0$, set

$$\sigma^{(\mu+1)}(X) = \sigma^{(\mu)}(X)$$

If $d_{\mu} \neq 0$, find $\rho < \mu$ such that $d_{\rho} \neq 0$ and $\rho - \ell_{\mu}$ is the largest. Then

$$\sigma^{(\mu+1)}(X) = \sigma^{(\mu)}(X) + d_{\mu} d_{\rho}^{-1} X^{\mu-\rho} \sigma^{(\rho)}(X).$$

$$\sigma(X) = \sigma^{(2t)}(X).$$

Roots of $\sigma(X)$

Substituting the elements $1, \alpha, \alpha^2, \cdots, \alpha^{2^m-2}$ of $GF(s^m)$ into $\sigma(X)$ if $\sigma(\alpha^i) = 0$, then

$$\alpha^{2^m-1-i}$$

is an error location number.

Error Values

$$Z(X) = 1 + (S_1 + \sigma_1)\, X + (S_2 + \sigma_1\, S_1 + \sigma_s)\, X^2$$

$$+\cdots+ (S_\nu + \sigma_1\, S_{\nu-1} +\cdots+ \sigma_{\nu-1}\, S_1 + \sigma_\nu)\, X^\nu$$

Let $\sigma'(X)$ be the derivative of $\sigma(X)$. Then

$$e_{f_\ell} = \frac{Z(\beta_\ell^{-1})}{\beta_\ell^{-1}\ \sigma'(\beta_\ell^{-1})} = \frac{Z(\beta_\ell^{-1})}{\sum\limits_{\substack{i=1 \\ i\neq\ell}}^{\nu} (1 + \beta_i\ \beta_\ell^{-1})}$$

FOUIER TRANSFORM IN GF(2^m)

Let

$$\bar{c}(X) = c_0 + c_1 X + \cdots c_{n-1} X^{n-1}$$

be a polynomial of degree n - 1 or less with coefficients from GF(2^m). The Fourier transform of $\bar{c}(X)$ is the polynomial

$$\bar{C}(X) = C_0 + C_1 X + \cdots + C_{n-1} X^{n-1}$$

where the j-th spectral component is given by

$$C_j = \bar{c}(\alpha^j) = \sum_{i=0}^{n-1} c_i \alpha^{ij}$$

Note that the spectral components $C_0, C_1, \cdots, C_{n-1}$ are also symbols in GF(2^m).

INVERSE TRANSFORM

Given $\bar{C}(X)$, the polynomial $\bar{c}(X)$ can be determined by taking the inverse Fourier transform of $\bar{C}(X)$ with

$$c_i = \frac{1}{n \text{ modulo } 2} C(\alpha^i)$$

$$= - \sum_{j=0}^{n-1} C_j \, \alpha^{-ij}$$

The polynomials $\bar{c}(X)$ and $\bar{C}(X)$ form a transform pair.

PROPERTIES

(1) The j-th spectral component C_j is zero if and only if α^j is a root of $\bar{c}(X)$.

(2) The i-th component c_i of $\bar{c}(X)$ is zero if and only α^{-i} is a root of $\bar{C}(X)$.

CHARACTERIZATION OF RS CODES IN FREQUENCY DOMAIN

Let $n = 2^m - 1$. Let $\bar{c}(X)$ be a code polynomial in the t-symbol-correcting primitive RS code whose generator polynomial has $\alpha, \alpha^2, \cdots, \alpha^{2t}$ as all its roots. Clearly the Fourier transform $\bar{C}(X)$ of $\bar{c}(X)$ has zero spectral components at positions $j = 1, 2, \cdots, 2t$, i.e.

$$C_1 = C_2 = \cdots\cdot = C_{2t} = 0$$

RS CODES IN FREQUENCY DOMAIN

In frequency domain, a primitive t-symbol-correcting RS code with symbols from GF(2^m) consists of all the polynomials

$$\bar{C}(X) = C_0 + C_1 X + \cdots + C_{n-1} X^{n-1}$$

over GF(2^m) for which

$$C_1 = C_2 = \cdots = C_{2t} = 0.$$

SYNDROME COMPUTATION

Let $\bar{r}(X) = r_0 + r_1 X + \cdots + r_{n-1} X^{n-1}$ be a received polynomial. Then

$$\bar{r}(X) = \bar{c}(X) + \bar{e}(X)$$

where $\bar{c}(X)$ and $\bar{e}(X)$ are the transmitted code polynomial and the error polynomial respectively. The Fourier transform of $\bar{r}(X)$ is

$$\bar{R}(X) = R_0 + R_1 X + \cdots + R_{n-1} X^{n-1}$$

with

$$R_j = \bar{r}(\alpha^j) = \sum_{i=0}^{n-1} r_i \alpha^{ij}$$

Since $\bar{R}(X) = \bar{C}(X) + \bar{E}(X)$,

$$R_j = C_j + E_j$$

where $\bar{C}(X)$ and $\bar{E}(X)$ are the Fourier transforms of $\bar{c}(X)$ and $\bar{e}(X)$ respectively. For $j = 1, 2, \cdots, 2t$,

$$C_j = 0$$

and

$$R_j = E_j.$$

SYNDROME (cont.)

Let $\bar{S} = (S_1, S_2, \cdots, S_{2t})$ be the syndrome of $\bar{r}(X)$. Then

$$S_j = \bar{r}(\alpha^j)$$

Thus, for $j = 1, 2, \cdots, 2t$,

$$S_j = R_j = E_j.$$

This says that the 2t spectral components $R_1, R_2, \cdots, R_{2t}$ of $\bar{R}(X)$ are the 2t syndrome components, and are equal to the 2t spectral components $E_1, E_2, \cdots, E_{2t}$ of $E(X)$, the transform of the error polynomial $\bar{e}(X)$.

DECODING

Once $S_1, S_2, \cdots, S_{2t}$ are computed, we may use Berlekamp's iterative method to determine the error location polynomial

$$\sigma(X) = \sigma_0 + \sigma_1 X + \cdots + \sigma_t X^t.$$

The 2t spectral components $E_1, E_2, \cdots, E_{2t}$ of $\bar{E}(X)$ are known, the other $n - 2t$ spectral components of $\bar{E}(X)$ can be computed from the following recursive equation:

$$E_{j+t} = \sigma_1 E_{j+t-1} + \sigma_2 E_{j+t-2} + \cdots + \sigma_t E_j$$

for $j = t + 1, t + 2, \cdots, n - 1 - t$. The component E_0 is given by

$$E_0 = \frac{1}{\sigma_\nu} (E_\nu + \sigma_1 E_{\nu-1} + \cdots + \sigma_{\nu-1} E_1)$$

where σ_ν is the coefficient of the highest power of $\sigma(X)$ that is not zero.

Once $E(X)$ is found, we take the inverse transform of

$$\bar{C}(X) = \bar{R}(X) - E(X).$$

This gives $\bar{c}(X)$.

DECODING

(1) Take the transform $\bar{R}(X)$ of $\bar{r}(X)$.

(2) Find $\sigma(X)$.

(3) Compute $\bar{E}(X)$.

(4) Take the inverse transform $\bar{C}(X)$ of $\bar{C}(X) = \bar{R}(X) - \bar{E}(X)$.

GALOIS FIELD

Consider the Galois field $GF(2^m)$. Let β be an element in $GF(2^m)$. The trace of β, denoted $Tr(\beta)$, is defined as

$$Tr(\beta) \triangleq \sum_{\iota=0}^{m-1} \beta^{2^{\iota}}$$

<u>Properties</u>:

(1) $Tr(\beta) \varepsilon GF(2)$.

(2) $Tr(\beta + \gamma) = Tr(\beta) + Tr(\gamma)$.

(3) For $a \varepsilon GF(2)$, $Tr(a\beta) = a\, Tr(\beta)$.

BASIS

$GF(2^m)$ may be regarded as an m-dimensional vector space over GF(2). For any factor λ of m, $GF(2^m)$ contains $GF(2^\lambda)$ as a subfield. Let β be an element not contained in any subfield of $GF(2^m)$. Then

$$\{\beta^0, \beta^1, \beta^2, \cdots, \beta^{m-1}\}$$

is a basis of $GF(2^m)$.

$\boxed{\text{DUAL BASIS}}$

Let $\{\ell_0, \ell_1, \cdots, \ell_{m-1}\}$ be expressed as,

$$z = a_0 \ell_0 + a_1 \ell_1 + \cdots + a_{m-1} \ell_{m-1}$$

where $a_i \varepsilon GF(2)$. The basis $\{\ell_0, \ell_1, \cdots, \ell_{m-1}\}$ is called the dual basis of $\{\beta^0, \beta^1, \beta^2, \cdots, \beta^{m-1}\}$ if

$$Tr(\beta^i \ell_j) = \begin{cases} 0 & \text{for} \quad i \neq j \\ 1 & \text{for} \quad i = j \end{cases}$$

with $0 \leq i, j < m$. In this case

$$a_i = Tr(\beta^i a).$$

BIT - SERIAL MULTIPLICATION

Consider the multiplication of an arbitrary element z by a fixed element G in GF(2^m). Let us express z and z · G in terms of the dual basis $\{\ell_0, \ell_1, \cdots, \ell_{m-1}\}$,

$$z = z_0 \ell_0 + z_1 \ell_1 + \cdots + z_{m-1} \ell_{m-1}$$

$$z \cdot G = z_0' \ell_0 + z_1' \ell_1 + \cdots + z_{m-1}' \ell_{m-1}$$

where

$$z_i = \mathrm{Tr}(\beta^i z)$$

$$z_i' = \mathrm{Tr}(\beta^i z G).$$

BIT - SERIAL MULTIPLICATION CONTINUED

The coefficient z'_i is related to z_0, $z_1, \cdots, z_{m-1}$ in linear form. First, we note that

$$z'_i = Tr(\beta^i \ z \ G)$$

$$= z_0 \ Tr(\beta^i \ G \ \ell_0) + z_1 \ Tr(\beta^i \ G \ \ell_1) + \cdots + z_{m-1} \ Tr(\beta^i \ G \ \ell_{m-1})$$

For $i = 0$, we have

$$z'_0 = z_0 \ Tr(G \ \ell_0) + z_1 \ Tr(G \ \ell_1) + \cdots + z_{m-1} \ Tr(G \ \ell_{m-1})$$

This says that z'_0 is simply a modulo-2 sum of the bits in $(z_0, z_1, \cdots, z_{m-1})$. The coefficients, $Tr(G \ \ell_i)$'s can be pre-determined.

BIT - SERIAL MULTIPLICATION CONTINUED

The coefficients z'_1, $z'_2, \cdots$, z'_{m-1} can be computed in a serial manner. Consider

$$\beta\, z = a_0\, \ell_0 + a_1\, \ell_1 + \cdots + a_{m-1}\, \ell_{m-1}$$

where

$$a_i = Tr[\beta^i (\beta z)]$$

$$= Tr[\beta^{i+1}\, z] = z_{i+1}$$

Hence

$$\beta\, z = z_1\, \ell_0 + z_2\, \ell_1 + \cdots + z_{m-1}\, \ell_{m-2} + z_m\, \ell_{m-1}$$

where

$$z_m = Tr(\beta^m\, z)$$

$$= z_0\, Tr(\beta^m\, \ell_0) + z_1\, Tr(\beta^m\, \ell_1) + \cdots + z_{m-1}\, Tr(\beta^m\, \ell_{m-1})$$

BIT - SERIAL MULTIPLICATION CONTINUED

Now

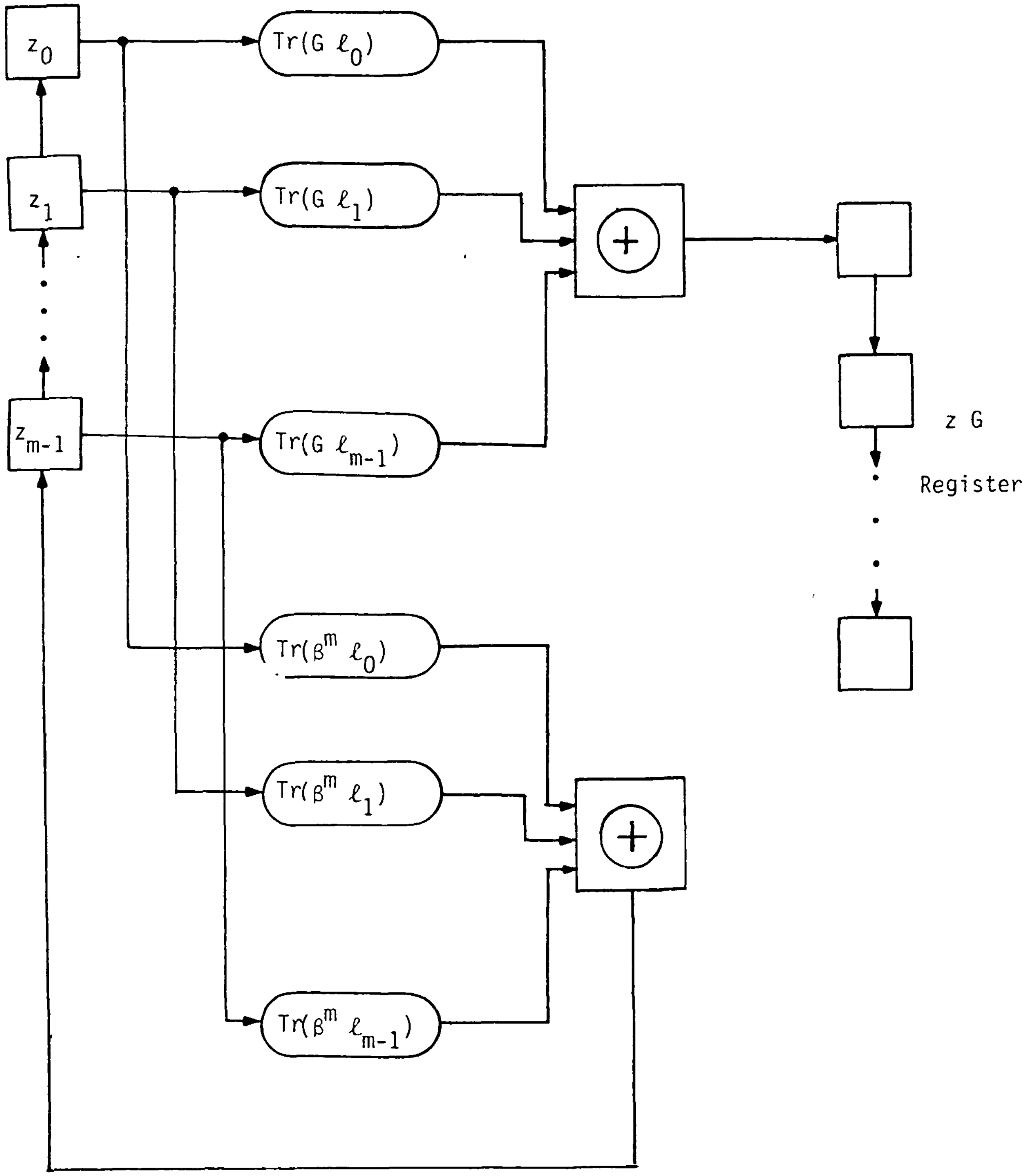
$$z_1' = Tr(\beta\ z\ G)$$

$$= z_1\ Tr\ (G\ \ell_0) + z_2\ Tr(G\ \ell_1) + \cdots + z_{m-1}\ Tr(G\ \ell_{m-2})$$

$$+ z_m Tr\ (G\ \ell_{m-1})$$

From the above expression, we see that, to generate z_1', we simply replace the vector $(z_0, z_1, \cdots, z_{m-1})$ by $(z_1, z_2, \cdots, z_m)$. This can be implemented easily by a sequential circuit as shown in next view graph. The z-register initially stores the representation of z in dual basis, $(z_0, z_1, \cdots, z_{m-1})$. After a total of m-shifts, z G is stored in z G-register.

BIT - SERIAL - MULTIPLIER



REVERSIBLE REED-SOLOMON CODES

Let γ be a primitive element in GF(2^m). Choose b such that

$$\gamma^{b} \cdot \gamma^{b+2t-1} = \gamma^{2^m-1} \qquad (1)$$

Consider the Reed-Solomon code whose generator polynomial $\bar{g}(X)$ has

$$\gamma^{b}, \gamma^{b+1}, \cdots, \gamma^{b+2t-1}$$

as all its roots. Then

$$\bar{g}(X) = \sum_{j=0}^{2t-1} (X + \gamma^{b+j})$$

$$= G_0 + G_1 X + \cdots + G_{2t} X^{2t}$$

The condition (1) gives the following property:

$$\begin{aligned} G_0 &= G_{2t} = 1 \\ G_1 &= G_{2t-1} \\ &\vdots \\ G_{t-1} &= G_{t+1}. \end{aligned}$$

Thus $\bar{g}(X)$ is self reciprol, and the RS code generated by $\bar{g}(X)$ is reversible.

ENCODING

The encoder for a reversible 2t-symbol-error-correction RS code requires at most t multiplers. The encoder can be implemented in terms of dual basis using bit-serial multipliers. Suppose that an information symbol

$$z = y_0 \alpha^0 + y_1 \alpha^1 + \cdots + y_{m-1} \alpha^{m-1}$$

is to be shifted into the encoder. It is first tranformed into the dual form,

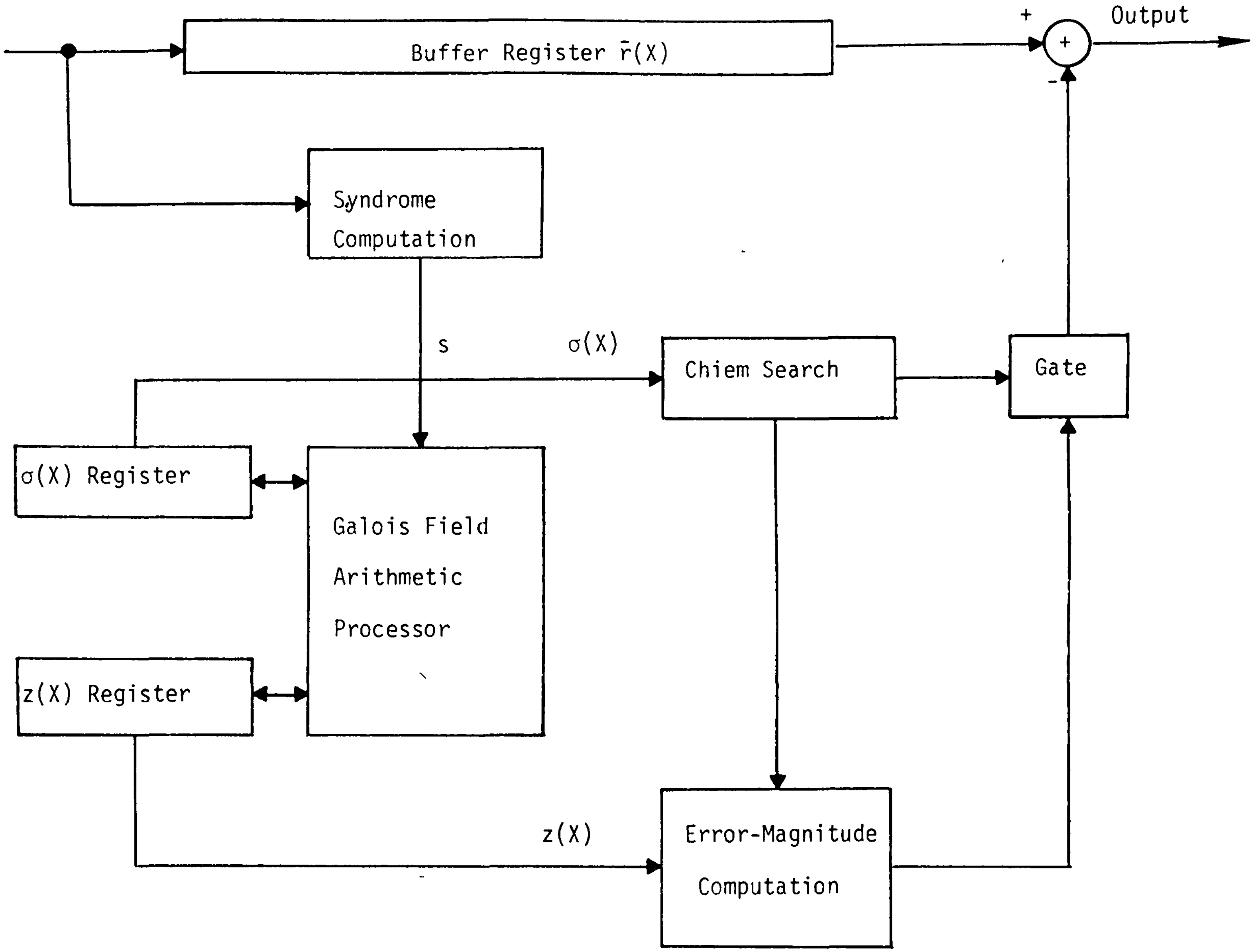
$$z = z_0 \ell_0 + z_1 \ell_1 + \cdots + z_{m-1} \ell_{m-1},$$

using the dual basis $\{\ell_0, \ell_1, \cdots, \ell_{m-1}\}$.

Then z is shifted into the encoder. The products

$$z\ G_0,\ z\ G_1, \cdots,\ z\ G_t$$

are formed using BS multipliers. It takes m clock times to form these products. Shift the encoder once. The next information symbol is ready to be shifted into the encoder.


Buffer Register $\bar{r}(X)$
+
−
Output
Syndrome Computation
s
σ(X)
Chiem Search
Gate
σ(X) Register
Galois Field Arithmetic Processor
z(X) Register
z(X)
Error-Magnitude Computation

PART II

REED SOLOMON CODES FOR ERROR CONTROL IN BYTE ORGANIZED COMPUTER MEMORY SYSTEMS

by

Huijie Deng and Daniel J. Costello, Jr.

Department of Electrical & Computer Engineering
Illinois Institute of Technology
Chicago, Illinois 60616
(312) 567-3404

July, 1984

Submitted to the IEEE Transactions on Computers

* This work was supported by NASA under Grant NAG 2-202.

ABSTRACT

A problem in designing semiconductor memories is to provide some measure of error control without requiring excessive coding overhead or decoding time. In LSI and VLSI technology, memories are often organized on a multiple bit (or byte) per chip basis. For example, some 256K-bit DRAM's are organized in 32K×8 bit-bytes. Byte oriented codes such as Reed Solomon (RS) codes can provide efficient low overhead error control for such memories. However, the standard iterative algorithm for decoding RS codes is too slow for these applications.

In this paper we present some special decoding techniques for extended single-and-double-error-correcting RS codes which are capable of high speed operation. These techniques are designed to find the error locations and the error values directly from the syndrome without having to use the iterative algorithm to find the error locator polynomial. Two codes are considered: 1) a $d_{min} = 4$ single-byte-error-correcting (SBEC), double-byte-error-detecting (DBED) RS code; and 2) a $d_{min} = 6$ double-byte-error-correcting (DBEC), triple-byte-error-detecting (TBED) RS code.

Index Terms

1. Error Control Coding

2. Error-Correcting-Codes

3. Error-Detecting-Codes

4. Reed-Solomon Codes

5. Byte-Oriented Codes

6. High-Speed Decoding

7. Computer Memory Systems

8. Byte Organized Chips

9. Chip Reliability

I. INTRODUCTION

Error control has long been used to improve the reliability of computer memory systems [1]. The most common approach has been to use a variation of the Hamming codes such as the single-error-correcting and double-error-detecting (SEC-DED) binary codes first introduced by Hsaio [2]. These codes are particularly effective for correcting and detecting errors in memories with a 1 bit per chip organization. In these memories a single chip failure can affect at most one bit in a codeword.

Large scale integration (LSI) and very large scale integration (VLSI) memory systems offer significant advantages in size, speed, and weight over earlier memory systems. These memories are normally packaged with a multiple bit (or byte) per chip organization. For example, some 256K-bit dynamic random access memories (DRAM's) are organized in 32K×8 bit-bytes. In this case a single chip failure can affect several or all of the bits in a byte, thus exceeding the error-correcting and detecting capability of SEC-DED codes.

Several papers have been written recently trying to extend the SEC-DED codes to include byte errors [3-7]. In this paper we investigate the use of Reed-Solomon (RS) codes for correcting and detecting byte errors in computer memories. RS codes are a class of nonbinary codes with symbols in the Galois field of 2^m elements ($GF(2^m)$). These codes are maximum distance separable (MDS), and thus can provide efficient low overhead error control for byte-organized memories, since symbol error correction in $GF(2^m)$ is equivalent to correcting an m-bit byte.

For computer memory applications, decoding must be fast and efficient. The standard approach to decoding RS codes uses the iterative algorithm [8] to form an error locator polynomial and then solves for its roots. It has the advantage of being easy to implement, but decoding is too slow for computer memory applications. High-speed decoding can be achieved by using the table-lookup method [1]. However, even for moderate code lengths, the implementation of table-lookup decoding is impractical, since either a large amount of storage or very complex logical circuitry is needed.

In this paper we investigate some special high-speed decoding techniques for extended single-and-double-byte-error-correcting RS codes. These techniques are designed to locate and correct the errors directly without having to use the iterative algorithm to find the error locator polynomial. Thus they satisfy the requirement of being both high-speed and easy to implement.

II. A $d_{min} = 4$ SBEC-DBED CODE

In this section we present an extended Reed-Solomon (RS) code over $GF(2^m)$ with minimum distance $d_{min} = 4$. This code can be used to correct any single byte error and simultaneously detect any double byte error. Thus it is called a single-byte-error-correcting (SBEC), double-byte-error-detecting (DBED) code. Fast encoding and decoding can be achieved due to some nice features of the code described below.

The $d_{min} = 4$ Extended RS Code and Its Properties

It has been shown [9] that there exists an $(n+3,n)$ $d_{min} = 4$

extended RS code over GF(2^m) with parity-check matrix given by

$$\underline{H} = [\ \underline{I}_{3\times 3} \ \vdots\ \underline{H}_1\ ], \qquad (1)$$

where $\underline{I}_{3\times 3}$ is the 3×3 identity matrix,

$$\underline{H}_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & & 1 \\ 1 & \alpha & \alpha^2 & \alpha^3 & \cdots & \alpha^{n-1} \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \cdots & \alpha^{2n-2} \end{bmatrix}, \qquad (2)$$

α is a primitive element of GF(2^m), and $n \leq 2^m-1$. Because $d_{min} = 4$, the code can be used for correcting any single byte error and simultaneously detecting any double byte errors [1].

From (1) and (2) we see that the $\underline{H}$ matrix has the following important properties.

1) $\underline{H}$ is in systematic form. Hence $\underline{G}$ - the generator matrix - is also in systematic form:

$$\underline{G} = [\ \underline{H}_1{}^T \ \vdots\ \underline{I}\ ], \qquad (3)$$

where $\underline{H}_1{}^T$ is the transpose of $\underline{H}_1$, and where $\underline{I}$ is the n×n identity matrix. This implies that encoding and decoding can be implemented in parallel.

2) The first nonzero element of every column of $\underline{H}$ is the unit element $\alpha^0 = 1$. (The advantage of this property will be seen later.)

3) For a systematic code with $d_{min} = d$, each column of $\underline{H}_1$ must contain at least d-1 nonzero elements. In (2), each column of $\underline{H}_1$ contains exactly $d-1 = 4-1 = 3$

nonzero elements. So $\underline{H}$ contains the minimum possible number of nonzero elements.

4) The number of nonzero elements in each row of $\underline{H}$ is equal.

Properties 3) and 4) simplify the implementation of the encoder and the decoder.

Error Correction and Error Detection

Let $\underline{v} = (v_0, v_1, \cdots, v_{n+2})$ be a code vector that is written into memory. Let $\underline{r} = (r_0, r_1, \cdots, r_{n+2})$ be the corresponding (possibly noisy) vector that is read from memory. Because of possible chip failures, $\underline{r}$ may be different from $\underline{v}$. The modulo-2 vector sum

$$\underline{e} = \underline{r} + \underline{v} = (e_0, e_1, \cdots, e_{n+2}), \tag{4}$$

where $e_i \neq 0$ for $r_i \neq v_i$ and $e_i = 0$ for $r_i = v_i$, is called the error pattern. When $\underline{r}$ is read, the decoder computes the syndrome $\underline{s}$,

$$\underline{s}^T = \underline{r}\ \underline{H}^T = (\underline{v} + \underline{e})\underline{H}^T = (s_0, s_1, s_2). \tag{5}$$

Since $\underline{v}\ \underline{H}^T = \underline{0}$, the syndrome $\underline{s}$, computed from the vector $\underline{r}$, depends only on the error pattern $\underline{e}$, and not on the transmitted code vector $\underline{v}$.

a) Single byte error correction

Let $\underline{s}_s$ denote the syndrome corresponding to a single byte error. Then from (5) we have

$$\underline{s}_s = e_i \underline{h}_i = \begin{bmatrix} s_0 \\ s_1 \\ s_2 \end{bmatrix}, \tag{6}$$

where e_i is the error value at location i, and h_i is the $i^{\underline{th}}$ column of $\underline{H}$, $0 \leq i \leq n+2$. Note that the first nonzero element of every column of $\underline{H}$ is the unit element α^0, and $e_i\alpha^0 = e_i$. Therefore the error value e_i is given directly by the first nonzero element of the syndrome.

The problem of locating the error is reduced to finding a column $\underline{h}_i$ of $\underline{H}$ which satisfies (6) (see Chien [10]). This can be done in the following way. Check the elements of the syndrome $\underline{s}$ to see

1) if $s_0 \neq 0$, $s_1 = s_2 = 0$, then $i = 0$,
2) if $s_1 \neq 0$, $s_0 = s_2 = 0$, then $i = 1$,
3) if $s_2 \neq 0$, $s_0 = s_1 = 0$, then $i = 2$.

Otherwise, since

$$\underline{s}_s = e_i\underline{h}_i = e_i \begin{bmatrix} 1 \\ \alpha^{i-3} \\ \alpha^{2(i-3)} \end{bmatrix} = \begin{bmatrix} s_0 \\ s_1 \\ s_2 \end{bmatrix}, \tag{6'}$$

$$\text{for } 3 \leq i \leq n+2,$$

we have

$$\alpha^{i-3} = \frac{s_1}{s_0} = \frac{s_2}{s_1}, \tag{7}$$

and i gives the error location. Define

$$u \triangleq s_1^{\,2} + s_0s_2, \tag{8}$$

and note that (7) is equivalent to

$$u = 0 \quad \text{for} \quad s_j \neq 0 \quad , \quad j\varepsilon\{0, 1, 2\}. \tag{9}$$

b) Double byte error detection

Let $\underline{s}_d$ denote the syndrome corresponding to a double byte error. Then from (5) we have

$$\underline{s}_d = e_j\underline{h}_j + e_k\underline{h}_k \ , \tag{10}$$

where $0 \leq j < k \leq n+2$.

The following theorem regarding the $\underline{H}$ matrix of a binary block code still holds true in the case of a nonbinary code (see [1]).

Theorem 1. A code defined by a parity-check matrix $\underline{H}$ will have minimum distance d if and only if every combination of d-1 or fewer columns of $\underline{H}$ is linearly independent.

Theorem 1 can be used directly to obtain the following property of the $d_{min} = 4$ extended RS code.

Property 1.

$$\underline{s}_s \neq \underline{s}_d \ , \tag{11}$$

for any single and double byte errors.

By property 1, double byte error detection can be done in the following way. If

$$s_{i1} = 0, \ s_{i2} \neq 0, \ s_{i3} \neq 0, \quad \text{where} \quad i_1, i_2, i_3 \varepsilon \{0,1,2\}, \tag{12}$$

or if

$$s_i \neq 0, \quad \text{for} \quad i = 0,\ 1,\ 2, \quad \text{and} \quad \frac{s_1}{s_0} \neq \frac{s_2}{s_1} \ , \tag{13}$$

then two or more byte errors are detected. Note that (13) implies that

$$u = s_1{}^2 + s_0 s_2 \neq 0. \tag{14}$$

Summarizing the above discussion, we have the following decoding scheme for the SBEC-DBED code defined by (1). Read $\underline{r}$, and compute the syndrome $\underline{s}^T = \underline{r}\ \underline{H}^T = (s_0, s_1, s_2)$. Let $w(\underline{s})$ denote the Hamming weight of the syndrome.

1) If $w(\underline{s}) = 0$, decide that no errors occurred.

2) If $w(\underline{s}) = 1$, then check:

   (i) If $s_0 \neq 0$, $e_i = s_0$, $i = 0$;

   (ii) If $s_1 \neq 0$, $e_i = s_1$, $i = 1$;

   (iii) If $s_2 \neq 0$, $e_i = s_2$, $i = 2$;

   where e_i gives the error value and i gives the error location.

3) If $w(\underline{s}) = 2$, decide that two or more byte errors occurred.

4) If $w(\underline{s}) = 3$, compute $u = s_1{}^2 + s_0 s_2$. If $u = 0$, calculate $\alpha^{i-3} = s_1/s_0$ and correct a single byte error with error value $e_i = s_0$ at location i. Otherwise, decide that two or more byte errors occurred.

III. A $d_{min} = 6$ DBEC-TBED CODE

In this section we first present a special high speed decoding technique for the double-byte-error-correcting (DBEC) and triple-byte-error-detecting (TBED) RS code with $d_{min} = 6$. Then a slightly modified technique is applied to decoding the extended RS code with two extra information symbols.

The d_{min} = 6 RS Code and Its Properties

The generator polynomial for the d_{min} = 6 RS code is given by

$$g(x) = \sum_{i=-2}^{2} (x+\alpha^{i}), \tag{15}$$

where α is a primitive element of $GF(2^{m})$. The parity-check matrix, $\underline{H}_2$, of the code specified by (15) can be written as

$$\underline{H}_2 = \begin{bmatrix} 1 & \alpha^{-2} & (\alpha^{-2})^{2} & \cdots & (\alpha^{-2})^{n-1} \\ 1 & \alpha^{-1} & (\alpha^{-1})^{2} & \cdots & (\alpha^{-1})^{n-1} \\ 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha & (\alpha)^{2} & \cdots & (\alpha)^{n-1} \\ 1 & \alpha^{2} & (\alpha^{2})^{2} & \cdots & (\alpha^{2})^{n-1} \end{bmatrix}, \tag{16}$$

where $n \leq 2^{m}-1$. Because the code has d_{min} = 6, then from theorem 1 every combination of d_{min} -1 = 5 or fewer columns of $\underline{H}_2$ is linearly independent, and the code is capable of correcting any two or fewer byte errors and simultaneously detecting any combination of three byte errors [1].

When $\underline{r} = \underline{v} + \underline{e}$ is read, the decoder computes the syndrome $\underline{s}$,

$$\underline{s}^{T} = \underline{r}\ \underline{H}_2^{T} = (\underline{v}+\underline{e})\underline{H}_2^{T} = \underline{e}\ \underline{H}_2^{T}$$

$$= (s_{-2}, s_{-1}, s_{0}, s_{1}, s_{2}). \tag{17}$$

Let $\underline{s}_s$, $\underline{s}_d$, and $\underline{s}_t$ denote the syndromes corresponding to single, double, and triple byte error patterns, respectively. Then from (17) we have:

$$\underline{s}_s = \begin{bmatrix} e_i\alpha^{-2i} \\ e_i\alpha^{-i} \\ e_i \\ e_i\alpha^{i} \\ e_i\alpha^{2i} \end{bmatrix}, \tag{18}$$

where e_i is the error value and i is the error location, $0 \leq i \leq n-1$,

$$\underline{s}_d = \begin{bmatrix} e_i\alpha^{-2i} + e_j\alpha^{-2j} \\ e_i\alpha^{-i} + e_j\alpha^{-j} \\ e_i + e_j \\ e_i\alpha^{i} + e_j\alpha^{j} \\ e_i\alpha^{2i} + e_j\alpha^{2j} \end{bmatrix}, \tag{19}$$

where $0 \leq i < j \leq n-1$, and

$$\underline{s}_t = \begin{bmatrix} e_i\alpha^{-2i} + e_j\alpha^{-2j} + e_k\alpha^{-2k} \\ e_i\alpha^{-i} + e_j\alpha^{-j} + e_k\alpha^{-k} \\ e_i + e_j + e_k \\ e_i\alpha^{i} + e_j\alpha^{j} + e_k\alpha^{k} \\ e_i\alpha^{2i} + e_j\alpha^{2j} + e_k\alpha^{2k} \end{bmatrix}, \tag{20}$$

where $0 \leq i < j < k \leq n-1$.

Before proceeding, we need to prove some properties of the $d_{min} = 6$ RS code which will be used later. Theorem 1 can be used directly to obtain property 1.

Property 1.

$$\underline{s}_s \neq \underline{s}_d \neq \underline{s}_t \, , \tag{21}$$

for any single, double, and triple byte errors.

Property 2. If α is a primitive element of $GF(2^m)$, then

$$\alpha^{-i} + \alpha^{-j} \neq 0, \tag{22.1}$$

$$\alpha^{-2i} + \alpha^{-2j} \neq 0, \tag{22.2}$$

for $0 \leq i < j \leq 2^m-2$.

Proof. If $\alpha^{-i} + \alpha^{-j} = 0$, multiply both sides by $\alpha^{i+j} \neq 0$. Then we have $\alpha^i + \alpha^j = 0$. But this is impossible since α is a primitive element. Similarly we can show that (22.2) is also correct. Q.E.D.

Let $\underline{s}_d = (s_{-2}, s_{-1}, s_0, s_1, s_2)^T$. From (19) we have the following equations:

$$s_{-2} = e_i\alpha^{-2i} + e_j\alpha^{-2j} \tag{23.1}$$

$$s_{-1} = e_i\alpha^{-i} + e_j\alpha^{-j} \tag{23.2}$$

$$s_0 = e_i + e_j \tag{23.3}$$

$$s_1 = e_i\alpha^i + e_j\alpha^j \tag{23.4}$$

$$s_2 = e_i\alpha^{2i} + e_j\alpha^{2j} \tag{23.5}$$

Property 3. Let $\underline{s}_d = (s_{-2}, s_{-1}, s_0, s_1, s_2)^T$ be the syndrome corresponding to a double byte error with error values e_i and e_j at locations i and j, respectively. Let N denote the number of zero elements of $\underline{s}_d$. Then

$$N \leq 2, \tag{24}$$

and the equal sign holds for some values of i and j in only two cases:

1) $s_{-1} = s_2 = 0$;
2) $s_1 = s_{-2} = 0$.

Proof: It can be seen from property 2 that the vectors $(\alpha^{-2i}, \alpha^{-2j})$, $(\alpha^{-i}, \alpha^{-j})$, $(1, 1)$, (α^{i}, α^{j}) and $(\alpha^{2i}, \alpha^{2j})$, where $0 \leq i < j \leq 2^m-2$, are always pairwise linearly independent except for the following two pairs:

1) $(\alpha^{-i}, \alpha^{-j})$, $(\alpha^{2i}, \alpha^{2j})$;
2) (α^{i}, α^{j}), $(\alpha^{-2i}, \alpha^{-2j})$.

These two pairs are linearly dependent for some values of i and j.

First we show that if $s_0 = 0$, then $s_k \neq 0$, $k = -2, -1, 1, 2$. Suppose $s_k = 0$ for some $k \neq 0$. From (23.1)-(23.5), we have

$$\begin{bmatrix} s_0 \\ s_k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = e_i \begin{bmatrix} 1 \\ \alpha^{ki} \end{bmatrix} + e_j \begin{bmatrix} 1 \\ \alpha^{kj} \end{bmatrix},$$

where $e_i \neq 0$, $e_j \neq 0$, and $k = -2, -1, 1, 2$. But $(1, 1)$ and $(\alpha^{ki}, \alpha^{kj})$ are linearly independent, and this implies that the above equation is impossible. Hence $s_k \neq 0$, $k = -2, -1, 1, 2$.

Next we show that if $s_{-1} = 0$ (or $s_2 = 0$), then $s_k \neq 0$, $k = -2, 0, 1$, and s_2 (or s_{-1}) can be either zero or nonzero. It is easy to show that $s_k \neq 0$, $k = -2, 0, 1$, in the same way as above. Because $(\alpha^{-i}, \alpha^{-j})$ and $(\alpha^{2i}, \alpha^{2j})$ are linearly dependent for some i and j, there exists $\beta_1 \neq 0$, $\beta_2 \neq 0$, β_1, $\beta_2 \varepsilon GF(2^m)$, and some $i < j$, such that

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \beta_1 \begin{bmatrix} \alpha^{-i} \\ \alpha^{-2i} \end{bmatrix} + \beta_2 \begin{bmatrix} \alpha^{-j} \\ \alpha^{-2j} \end{bmatrix}.$$

Let $e_i = \beta_1$ and $e_j = \beta_2$. From (23.2) and (23.5) we see that the above equation becomes

$$\begin{bmatrix} s_{-1} \\ s_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = e_i \begin{bmatrix} \alpha^{-i} \\ \alpha^{-2i} \end{bmatrix} + e_j \begin{bmatrix} \alpha^{-j} \\ \alpha^{-2j} \end{bmatrix}.$$

Therefore $s_{-1} = s_2 = 0$ for some i and j.

By exactly the same argument as above, we can prove that if s_1 (or s_{-2}) = 0, then $s_k \neq 0$, $k = -1, 0, 2$, and s_{-2} (or s_1) can be either zero or nonzero. This completes the proof that $N \leqq 2$. Q.E.D.

Property 4. Let $\underline{s}_d = (s_{-2}, s_{-1}, s_0, s_1, s_2)^T$. Then

$$s_2 s_{-2} + s_0^2 \neq 0 \tag{25.1}$$

$$s_1 s_{-2} + s_{-1} s_0 \neq 0 \tag{25.2}$$

$$s_0 s_1 + s_2 s_{-1} \neq 0 \tag{25.3}$$

hold true for all double byte errors.

Proof: 1) Suppose $s_2 s_{-2} + s_0^2 = 0$. Them from (23.1), (23.3), and (23.5) we obtain

$$(e_j\alpha^{2i} + e_j\alpha^{2j})(e_j\alpha^{-2i} + e_j\alpha^{-2j}) + (e_i + e_j)^2 = 0.$$

Expanding this equation and performing some simplification give us

$$\alpha^{2i-2j} + \alpha^{-2i+2j} = 0.$$

But this is impossible, since α is a primitive element and $i \neq j$. Therefore, $s_2 s_{-2} + s_0{}^2 \neq 0$.

2) Suppose $s_1 s_{-2} + s_{-1} s_0 = 0$, i.e., $s_1 s_{-2} = s_{-1} s_0$. From (23.1)-(23.4) we have

$$(e_i \alpha^i + e_j \alpha^j)(e_i \alpha^{-2i} + e_j \alpha^{-2j}) = (e_i \alpha^{-i} + e_j \alpha^{-j})(e_i + e_j).$$

After some simplification we obtain

$$\alpha^{i-2j} + \alpha^{j-2i} = \alpha^{-i} + \alpha^{-j}.$$

Multiplying both sides by $\alpha^{2i+2j} \neq 0$, the above equation becomes

$$\alpha^{3i} + \alpha^{3j} = \alpha^{i+2j} + \alpha^{j+2i}, \qquad (26)$$

or

$$(\alpha^i + \alpha^j)(\alpha^{2i} + \alpha^{i+j} + \alpha^{2j}) = \alpha^{i+j}(\alpha^i + \alpha^j).$$

This can be reduced to

$$\alpha^{2i} + \alpha^{2j} = 0, \quad i \neq j.$$

But this is impossible. Hence $s_1 s_{-2} + s_{-1} s_0 \neq 0$.

3) Suppose $s_0 s_1 + s_2 s_{-1} = 0$. In the same way as above we obtain

$$\alpha^{3i} + \alpha^{3j} = \alpha^{i+2j} + \alpha^{j+2i}.$$

This is exactly the same as (26). Hence the equality is invalid, and $s_0 s_1 + s_2 s_{-1} \neq 0$. Q.E.D.

<u>Decoding Using the Quadratic Equation</u>

In this subsection we show that the well known quadratic equation over $GF(2^m)$ can be used to decode the $d_{min} = 6$ RS code. Also, we present a method for solving it.

It was shown in property 2 that if α is a primitive element of $GF(2^m)$, then $\alpha^{-i} + \alpha^{-j} \neq 0$ and $\alpha^{-2i} + \alpha^{-2j} \neq 0$ both hold true for any $0 \leq i < j \leq 2^m-2$. From (23.1) and (23.3) we have

$$e_i = \frac{\det\begin{vmatrix} s_0 & 1 \\ s_{-2} & \alpha^{-2j} \end{vmatrix}}{\det\begin{vmatrix} 1 & 1 \\ \alpha^{-2i} & \alpha^{-2j} \end{vmatrix}} = \frac{s_{-2} + s_0\alpha^{-2j}}{(\alpha^{-i} + \alpha^{-j})^2} . \tag{27}$$

From (23.2) and (23.3) we have

$$e_i = \frac{\det\begin{vmatrix} s_0 & 1 \\ s_{-1} & \alpha^{-j} \end{vmatrix}}{\det\begin{vmatrix} 1 & 1 \\ \alpha^{-i} & \alpha^{-j} \end{vmatrix}} = \frac{s_{-1} + s_0\alpha^{-j}}{\alpha^{-i} + \alpha^{-j}} . \tag{28}$$

Therefore

$$\frac{s_1 + s_0\alpha^{-j}}{\alpha^{-i} + \alpha^{-j}} = \frac{s_{-2} + s_0\alpha^{-2j}}{(\alpha^{-i} + \alpha^{-j})^2} . \tag{29}$$

Now multiplying both sides by $(\alpha^{-i} + \alpha^{-j})^2 \neq 0$, (29) becomes

$$(\alpha^{-i} + \alpha^{-j})(s_{-1} + s_0\alpha^{-j}) = s_{-2} + s_0\alpha^{-2j} . \tag{30}$$

After simplification we have

$$s_{-1}(\alpha^{-i} + \alpha^{-j}) + s_{-2} + s_0\alpha^{-i-j} = 0. \qquad (31)$$

Multiplying (31) by α^{i+j} gives us

$$s_{-1}(\alpha^i + \alpha^j) + s_{-2}\alpha^i\alpha^j + s_0 = 0. \qquad (32)$$

In the same way, from (23.3)-(23.5), we can obtain

$$s_1(\alpha^i + \alpha^j) + s_0\alpha^i\alpha^j + s_2 = 0. \qquad (33)$$

Now define

$$b \triangleq \alpha^i + \alpha^j, \qquad (34.1)$$

$$c \triangleq \alpha^i\alpha^j. \qquad (34.2)$$

Also define

$$\gamma_1 \triangleq s_0{}^2 + s_{-1}s_1, \qquad (35.1)$$

$$\gamma_2 \triangleq s_2s_{-2} + s_0{}^2, \qquad (35.2)$$

$$\gamma_3 \triangleq s_1s_{-2} + s_{-1}s_0, \qquad (35.3)$$

$$\gamma_4 \triangleq s_0s_1 + s_2s_{-1}. \qquad (35.4)$$

Solving (32) and (33) for $b = \alpha^i + \alpha^j$ and $c = \alpha^i\alpha^j$, we obtain

$$b = \alpha^i + \alpha^j = \frac{\gamma_2}{\gamma_3}, \qquad (36.1)$$

$$c = \alpha^i\alpha^j = \frac{\gamma_4}{\gamma_3}, \qquad (36.2)$$

for $\gamma_3 \neq 0$. Also, from (34.1) and (34.2) we see that α^i and α^j are the roots of

$$y^2 + by + c = 0. \qquad (37)$$

This is the well-known quadratic equation over GF(2^m). We will see later that (37) plays an important role in decoding.

Therefore we call it the "decoding equation". Because of its importance, in the remainder of this subsection we present a method of solving it.

The formula for the roots of the quadratic equation $y^2 + by + c = 0$ is $(-b \pm \sqrt{b^2 - 4c})/2$. Unfortunately, for finite fields of characteristic two, this formula is not applicable because the denominator is zero $(2 = 1 + 1 = 0)$. However, there are several known approaches to solving this problem. One way of finding the roots is by trying each element of the field in sequence [11]. But this is unacceptable for fast decoding because it takes a long time. The method given in [12] is probably the best one known. We present it here.

Define

$$x \triangleq y/b. \tag{38}$$

Then (37) becomes

$$x^2 + x + K = 0, \tag{39}$$

where

$$K \triangleq c/b^2. \tag{40}$$

Let β be any element of $GF(2^m)$, and define

$$T_2(\beta) \triangleq \sum_{i=0}^{m-1} \beta^{2^i} \tag{41}$$

$T_2(\beta)$ is known as the trace of β. It is either zero or one [12]. For even m, define

$$T_4(\beta) \triangleq \sum_{i=0}^{(m-2)/2} \beta^{2^{2i}} \quad , \quad m \text{ even.} \tag{42}$$

If (39)has solutions, $T_4(K)$ is either zero or one [12]. Equation (39) has solutions in $GF(2^m)$ if and only if $T_2(K) = 0$ [8,13].

Suppose $T_2(K) = 0$, i.e., (39) has solutions. Let x_1 be a solution of (39). Then $x_2 = 1 + x_1$ is the other solution. Then we have the following results [12]:

1) m odd

$$x_1 = \sum_{j\varepsilon J} K^{2^j} = \sum_{i\varepsilon I} K^{2^i} \qquad (43)$$

where $I = \{1,3,5, \cdots, m-2\}$, $J = \{0,2,4, \ldots, m-1\}$.

2) $m \equiv 2$ modulo 4

$$x_1 = \sum_{i=0}^{(m-6)/4} (K+K^2)^{2^{2+4i}} \quad , \quad \text{for } T_4(K) = 0, \quad (44.1)$$

$$x_1 = \alpha_1 + \sum_{i=0}^{(m-6)/4} (K+K^2)^{2^{2+4i}}, \quad \text{for } T_4(K) = 1, (44.2)$$

where α_1 is a solution of the equation $\alpha_1{}^2 + \alpha_1 + 1 = 0$.

3) $m \equiv 0$ modulo 4

$$x_1 = S + S^2 + K^{2^{m-1}} \left(1 + \sum_{i=0}^{(m/4)-1} K^{2^{2i+m/2}}\right) ,$$
$$\text{for } T_4(K) = 1, \qquad (45)$$

where

$$S = \sum_{j=1}^{(m/4)-1} \sum_{i=j}^{(m/4)-1} K^{(2^{2i-1+m/2} + 2^{2j-2})} .$$

For $T_4(K) = 0$, select an element β of $GF(2^m)$ such that $T_2(\beta) = 1$, compute $K_1 = \beta+\beta^2$, and solve $z^2 + z + K_1 + K = 0$ using (45) with K replaced by $K_1 + K$. Then $x_1 = \beta+z_1$ is a solution of (39), where z_1 is obtained from (45). For m = 4, 8, 12, (45) reduces to the following forms:

$$m = 4, \quad x_1 = K^8 + K^{12};$$

$$m = 8, \quad x_1 = K^{33} + K^{66} + K^{129} + K^{132};$$

$$m = 12, \quad x_1 = K^{2048}(1 + K^{64} + K^{256} + K^{1024}) + K^{129} + K^{258} + K^{513} + K^{1026} + K^{516} + K^{1032}.$$

Decoding of the $d_{min} = 6$ RS Code

Suppose that a double byte error with error values e_i and e_j at locations i and j (i < j) occurs. By our definition, $\underline{s}_d = (s_{-2}, s_{-1}, s_0, s_1, s_2)^T$ is the syndrome associated with this error pattern. From property 4 we know that

$$\gamma_2 = s_2 s_{-2} + s_0^2 \neq 0,$$

$$\gamma_3 = s_1 s_{-2} + s_{-1} s_0 \neq 0,$$

and $$\gamma_4 = s_0 s_1 + s_2 s_{-1} \neq 0.$$

Therefore b and c in (36.1) and (36.2) exist. Hence (37) has two roots, α^i and α^j. We summarize as a theorem.

Theorem 2. If $s_{-2}, s_{-1}, s_0, s_1, s_2$ are the elements of $\underline{s}_d$, the decoding equation (37) has two roots, α^i and α^j, where i and j are the two byte error locations and $0 \leq i < j \leq 2^m-2$. In other words, whenever a double byte error occurs, its error locations can be found by solving the decoding equation (37).

Since $\alpha^i + \alpha^j \neq 0$ when α is a primitive element of $GF(2^m)$, (23.3), (23.4), and (36.1) imply that

$$e_i = \frac{\det\begin{vmatrix} s_0 & 1 \\ s_1 & \alpha^j \end{vmatrix}}{\det\begin{vmatrix} 1 & 1 \\ \alpha^i & \alpha^j \end{vmatrix}} = \frac{s_0\alpha^j + s_1}{\alpha^i + \alpha^j} = \frac{s_0\alpha^j + s_1}{b}, \qquad (46.1)$$

and

$$e_j = s_0 + e_i, \qquad (46.2)$$

where e_i and e_j are the error values at locations i and j of the double byte error.

Now let $\underline{s}_s = (s_{-2}, s_{-1}, s_0, s_1, s_2)^T$ be the syndrome corresponding to a single byte error with error value e_i at location i. From (18) we have:

$$s_{-2} = e_i\alpha^{-2i} \qquad (47.1)$$

$$s_{-1} = e_i\alpha^{-i} \qquad (47.2)$$

$$s_0 = e_i \qquad (47.3)$$

$$s_1 = e_i\alpha^i \qquad (47.4)$$

$$s_2 = e_i\alpha^{2i}. \qquad (47.5)$$

From (47.1)-(47.5), we see that

$$s_i \neq 0, \quad \text{for} \quad i = -2, -1, 0, 1, 2, \qquad (48.1)$$

and

$$\frac{s_{-1}}{s_{-2}} = \frac{s_0}{s_{-1}} = \frac{s_1}{s_0} = \frac{s_2}{s_1} = \alpha^i. \qquad (48.2)$$

Note that (48.2) is equivalent to

$$\gamma_1 = s_0^{\,2} + s_{-1}s_1 = 0, \qquad (49.1)$$

$$\gamma_3 = s_1 s_{-2} + s_{-1} s_0 = 0 \tag{49.2}$$

$$\gamma_4 = s_0 s_1 + s_2 s_{-1} = 0. \tag{49.3}$$

The above result implies the following theorem.

Theorem 3. If s_{-2}, s_{-1}, s_0, s_1, s_2 are the elements of $\underline{s}_s$, then $s_i \neq 0$, for $i = -2, -1, 0, 1, 2$, and $\gamma_1 = \gamma_3 = \gamma_4 = 0$.

In other words, whenever a single byte error occurs, $s_i \neq 0$ for $i = -2, -1, 0, 1, 2$, and $\gamma_1 = \gamma_3 = \gamma_4 = 0$. From (47.3) and (47.4) we have

$$\alpha^i = \frac{s_1}{s_0}, \tag{50.1}$$

$$e_i = s_0, \tag{50.2}$$

where i gives the error location and e_i is the error value of a single byte error..

Properties 1-4 and theorems 2 and 3 imply the following theorem.

Theorem 4. If more than two elements of the syndrome $\underline{s} = (s_{-2}, s_{-1}, s_0, s_1, s_2)^T$ equal zero; or if γ_2, γ_3, γ_4 are not all equal to zero, but at least one of them does equal zero; or if the decoding equation (37) does not have roots in $GF(2^m)$; then at least three byte errors have occurred.

We now summarize the decoding scheme obtained above for the DBEC-TBED RS code defined by (15) and (16). Read $\underline{r}$, and calculate the syndrome $\underline{s}^T = \underline{r}\ \underline{H}_2^T = (s_{-2}, s_{-1}, s_0, s_1, s_2)$. Let $w(\underline{s})$, $w(\underline{\gamma}')$, and $w(\underline{\gamma}'')$ denote the Hamming weights of $\underline{s} = (s_{-2}, s_{-1}, s_0, s_1, s_2)^T$, $\underline{\gamma}' \triangleq (\gamma_1, \gamma_3, \gamma_4)$, and $\underline{\gamma}'' \triangleq (\gamma_2, \gamma_3, \gamma_4)$, respectively.

1) If $w(\underline{s}) = 0$, decide that no errors occurred.

2) If $1 \leqq w(\underline{s}) \leqq 2$, decide that at least three byte errors occurred.

3) If $3 \leq w(\underline{s}) \leq 4$, go to step 5

4) If $w(\underline{s}) = 5$, compute $\underline{\gamma}'$. If $w(\underline{\gamma}') = 0$, calculate $\alpha^i = \frac{s_1}{s_0}$, and correct a single byte error with error value $e_i = s_0$ at location i. If $w(\underline{\gamma}') \neq 0$, go to step 5.

5) Compute $\underline{\gamma}''$. If $w(\underline{\gamma}'') < 3$, or if $w(\underline{\gamma}'') = 3$ but $T_2(K) = 1$, decide that at least three byte errors occurred.

6) If $w(\underline{\gamma}'') = 3$ and $T_2(K) = 0$, solve the decoding equation (37) and find the roots α^i and α^j. Compute $e_i = (s_0\alpha^j + s_1)/b$ and $e_j = s_0 + e_i$, and correct a double byte error with error values e_i and e_j at locationsi and j, respectively.

Decoding of the Extended Code

The parity-check matrix $\underline{H}_2$ given in (16) can be extended to form a new parity-check matrix given by

$$\underline{H}_3 = \begin{bmatrix} & 1 & 0 \\ & 0 & 0 \\ \underline{H}_2 & 0 & 0 \\ & 0 & 0 \\ & 0 & 1 \end{bmatrix} . \tag{51}$$

The code specified by $\underline{H}_3$ is an $(n+2, n-3)d_{min} = 6$ extended RS code, where $n \leq 2^m-1$ [14, 15, 16].

It again follows from Theorem 1 that

$$\underline{s}_s \neq \underline{s}_d \neq \underline{s}_t \tag{52}$$

for all single, double, and triple byte errors. If the error locations are confined to locations 0 through n-1, all the previous results apply.

Now assume that errors occur at location n or n+1. Then the syndrome is given by:

$$\underline{s}_s = \begin{bmatrix} e_n \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} s_{-2} \\ s_{-1} \\ s_0 \\ s_1 \\ s_2 \end{bmatrix}, \tag{53.1}$$

for a single byte error at location n, or

$$\underline{s}_s = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ e_{n+1} \end{bmatrix} \begin{bmatrix} s_{-2} \\ s_{-1} \\ s_0 \\ s_1 \\ s_2 \end{bmatrix} \tag{53.2}$$

if the error is at location n+1. For a double byte error the syndrome is given by:

$$\underline{s}_d = \begin{bmatrix} e_i\alpha^{-2i} + e_n \\ e_i\alpha^{-i} \\ e_i \\ e_i\alpha^{i} \\ e_i\alpha^{2i} \end{bmatrix} = \begin{bmatrix} s_{-2} \\ s_{-1} \\ s_0 \\ s_1 \\ s_2 \end{bmatrix} \tag{54.1}$$

with two errors at locations i and n, respectively, where $0 \leq i \leq n-1$; and

$$\underline{s}_d = \begin{bmatrix} e_i\alpha^{-2i} \\ e_i\alpha^{-i} \\ e_i \\ e_i\alpha^{i} \\ e_i\alpha^{2i} + e_{n+1} \end{bmatrix} = \begin{bmatrix} s_{-2} \\ s_{-1} \\ s_0 \\ s_1 \\ s_2 \end{bmatrix}, \qquad (54.2)$$

with two errors at locations i and n+1, respectively, where $0 \leq i \leq n-1$. Finally

$$\underline{s}_d = \begin{bmatrix} e_n \\ 0 \\ 0 \\ 0 \\ e_{n+1} \end{bmatrix} = \begin{bmatrix} s_{-2} \\ s_{-1} \\ s_0 \\ s_1 \\ s_2 \end{bmatrix}, \qquad (54.3)$$

with two errors at locations n and n+1, respectively.

From (54.1)-(54.3) we obtain the following results.

1) If

$$s_{-2} \neq s_{-1} = s_0 = s_1 = s_2 = 0, \qquad (55)$$

then a single byte error occurred. From (54.1), we have the error value $e_n = s_{-2}$ and the error location n.

2) If

$$s_2 \neq s_{-2} = s_{-1} = s_0 = s_1 = 0, \qquad (56)$$

then a single byte error occured with error value $e_{n+1} = s_2$ at location n+1.

3) If

$$s_i \neq 0, \quad \text{for} \quad i = -1, 0, 1, 2, \qquad (57.1)$$

and

$$\frac{s_{-2}}{s_{-1}} \neq \frac{s_{-1}}{s_0} = \frac{s_0}{s_1} = \frac{s_1}{s_2}\ , \tag{57.2}$$

then a double byte error occurred. From (54.1) we have the error values $e_i = s_0$ and $e_n = s_{-2} + e_i\alpha^{-2i} = s_{-2} + s_0\alpha^{-2i}$ at locations i and n, respectively, where i is obtained from $\alpha^i = \frac{s_1}{s_0}$. Note that (57.2) is equivalent to

$$\gamma_1 = s_0^{\ 2} + s_{-1}s_1 = 0 \tag{58.1}$$

$$\gamma_3 = s_1s_{-2} + s_{-1}s_0 \neq 0 \tag{58.2}$$

$$\gamma_4 = s_0s_1 + s_2s_{-1} = 0 \tag{58.3}$$

4) If

$$s_i \neq 0, \quad \text{for} \quad i = -2, -1, 0, 1, \tag{59.1}$$

and

$$\frac{s_{-1}}{s_{-2}} = \frac{s_0}{s_{-1}} = \frac{s_1}{s_0} \neq \frac{s_2}{s_1}\ , \tag{59.2}$$

i.e.,

$$\gamma_1 = s_0^{\ 2} + s_{-1}s_1 = 0 \tag{60.1}$$

$$\gamma_3 = s_1s_{-2} + s_{-1}s_0 = 0 \tag{60.2}$$

$$\gamma_4 = s_0s_1 + s_2s_{-1} \neq 0 \tag{60.3}$$

then a double byte error occurred with error values $e_i = s_0$ and $e_{n+1} = s_2 + s_0\alpha^{2i}$ at locations i and n+1, respectively, where i is obtained from $\alpha^i = \frac{s_1}{s_0}$.

5) If

$$s_{-2} \neq 0, \; s_2 \neq 0, \quad \text{and} \quad s_{-1} = s_0 = s_1 = 0, \qquad (61)$$

then a double byte error occurred with error values $e_n = s_{-2}$ and $e_{n+1} = s_2$ at locations n and n+1, respectively.

Now we combine the discussion in this subsection with that of the previous subsections to obtain the following decoding scheme for the DBEC-TBED RS code defined by (51). From the vector $\underline{r}$, compute the syndrome $\underline{s}^T = \underline{r}\,\underline{H}_3^T = (s_{-2}, s_{-1}, s_0, s_1, s_2)$. Again let $w(\underline{s})$, $w(\underline{\gamma}')$, and $w(\underline{\gamma}'')$ denote the Hamming weights of $\underline{s} = (s_{-2}, s_{-1}, s_0, s_1, s_2)^T$, $\underline{\gamma}' = (\gamma_1, \gamma_3, \gamma_4)$, and $\underline{\gamma}'' = (\gamma_2, \gamma_3, \gamma_4)$, respectively.

1) If $w(\underline{s}) = 0$, decide that no errors occurred.

2) If $w(\underline{s}) = 1$, then check:

   (i). If $s_{-2} \neq 0$, correct a single byte error with error value $e_n = s_{-2}$ at location n;

   (ii). If $s_2 \neq 0$, correct a single byte error with error value $e_{n+1} = s_2$ at location n+1;

   (iii). Otherwise, decide that at least three byte errors occurred.

3) If $w(\underline{s}) = 2$, then check:

   (i). If $s_{-2} \neq 0$, $s_2 \neq 0$, correct two byte errors with error values $e_n = s_{-2}$ at $e_{n+1} = s_2$ at locations n and n+1, respectively.

   (ii). Otherwise, decide that at least three byte errors occurred.

4) If $w(\underline{s}) = 3$, go to step 7.

5) If $w(\underline{s}) = 4$, then check:

(i). If $s_{-2} = 0$, compute $\underline{\gamma}'$. If $w(\underline{\gamma}') = 1$ and $\gamma_3 \neq 0$, compute $\alpha^i = \frac{s_1}{s_0}$ and correct two byte errors with error values $e_i = s_0$ and $e_n = s_{-2} + s_0\alpha^{-2i}$ at locations i and n, respectively. If not, go to step 7.

(ii). If $s_2 = 0$, compute $\underline{\gamma}'$. If $w(\underline{\gamma}') = 1$ and $\gamma_4 \neq 0$, compute $\alpha^i = \frac{s_1}{s_0}$ and correct two byte errors with error values $e_i = s_0$ and $e_{n+1} = s_2 + s_0\alpha^{2i}$ at locations i and n+1, respectively. If not, go to step 7.

(iii). Otherwise, go to step 7.

6) If $w(\underline{s}) = 5$, compute $\underline{\gamma}'$. If $w(\underline{\gamma}') = 0$, compute $\alpha^i = \frac{s_1}{s_0}$ and correct a single byte error with error value $e_i = s_0$ at location i. If not, go to step 7.

7) Compute $\underline{\gamma}''$. If $w(\underline{\gamma}'') < 3$, or if $w(\underline{\gamma}'') = 3$ but $T_2(K) = 1$, decide that at least three byte errors occurred.

8) If $w(\underline{\gamma}'') = 3$ and $T_2(K) = 0$, solve the decoding equation (37) and find the roots α^i and α^j. Compute $e_i = (s_0\alpha^j + s_1)/b$, $e_j = s_0 + e_i$, and correct two byte errors values e_i and e_j at locations i and j, respectively.

IV. CONCLUSIONS

We have presented new decoding techniques for two byte oriented RS codes. These decoding techniques are based directly on the syndrome, and do not involve applying the iterative algorithm to find the error locator polynomial. Hence high-speed decoding can be achieved, making these codes well suited for error correction and detection in byte-organized computer memory systems such as LSI and VLSI chips.

The $d_{min} = 4$ code is capable of single-byte-error-correction (SBEC) and double-byte-error-detection (DBED) and can be extended to include three additional information symbols. The $d_{min} = 6$ code is capable of double-byte-error-correction (DBEC) and triple-byte-error-detection (TBED) and can be extended to include two additional information symbols. The decoding method applies to the extended codes with only slight modification.

Code efficiency is high since only three parity symbols are used in the $d_{min} = 4$ code and only five in the $d_{min} = 6$ code. In addition, the basic code length n can be selected to match the organization of the memory (as long as $n \leqq 2^m-1$) without changing the decoding method. However, efficiency is maximized when $n = 2^m-1$ is chosen.

REFERENCES

[1]. S. Lin and D.J. Costello, Jr., Error Control Coding: Fundamentals and Applications, Prentice-Hall, New Jersey, 1983.

[2]. M.Y. Hsiao, "A Class of Optimal Minimum Odd-Weight-Column SEC-DED Codes", IBM J. Res. Dev., 14, pp. 395-401, July 1970.

[3]. D.C.Bossen, L.C. Chang, and C.L. Chen, "Measurement and Generation of Error Correcting Codes for Package Failures", IEEE Trans. Comput., C-27, pp. 201-204, March 1978.

[4]. S.M. Reddy, "A Class of Linear Codes for Error Control in Byte-per-Card Organized Digital Systems", IEEE Trans. Comput., C-27, pp. 455-459, May 1978.

[5]. T.T. Dao, "SEC-DED Nonbinary Code for Fault-Tolerant Byte-Organized Memory Implemented with Quaternary Logic", IEEE Trans. Comput., C-30, pp. 662-666, Sept. 1981.

[6]. L.A. Dunning and M.R. Varanasi, "Code Constructions for Error Control in Byte Organized Memory Systems", IEEE Trans. Comput., C-32, pp. 535-542, June 1983.

[7]. C.L. Chen, "Error-Correcting Codes with Byte Error-Detection Capability", IEEE Trans. Comput., C-32, pp. 615-621, July 1983.

[8]. E.R. Berlekamp, Algebraic Coding Theory, McGraw-Hill, New York, 1968.

[9]. F.J. MacWilliams and N.J.A. Sloane, Theory of Error-Correcting Codes, North Holland, Amsterdam, 1978.

[10]. R.T. Chien, "Block-Coding Techniques for Reliable Data Transmission", IEEE Trans. on Comm. Tech., COM-19 (part II), pp. 743-751, Oct. 1971.

[11]. R.T. Chien, "Cyclic Decoding Procedures for BCH Codes", IEEE Trans. Inform. Theory, IT-10, pp. 357-363, Oct. 1964.

[12]. C.L. Chen, "Formulas for the Solutions of Quadratic Equations", IEEE Trans. Inform. Theory, IT-28, pp. 792-794, Sept. 1982.

[13]. E.R. Berlekamp, H. Ramsey, and G. Solomon, "On the Solution of Algebraic Equations Over Finite Fields", Inform. Contr., 18, pp. 553-564, Oct. 1967.

[14]. T. Kasami, S. Lin, and W.W. Peterson, "Some Results on Weight Distributions of BCH Codes", IEEE Trans. Inform. Theory, IT-12, p. 274, April 1966.

[15]. T. Kasami, S. Lin, and W.W. Peterson, "Some Results on Cyclic Codes Which are Invariant Under the Affine Group", Scientific Report AFCRL-66-662, Air Force Cambridge Research Labs., Bedford, MA, 1966.

[16]. J.K. Wolf, "Adding Two Information Symbols to Certain Nonbinary BCH Codes and Some Applications", Bell Sys. Tech. J., 48, pp. 2405-2424, 1969.