(NASA-CR-173619) AN EXTENDED d(MIN) = 4 RS CODE (Illinois Inst. of Tech.) 7 p HC A02/MF A01 CSCL 09B

N84-26327

Unclas
G3/61 19489

"AN EXTENDED d_{min} = 4 RS CODE"

by

H. Deng and Daniel J. Costello, Jr.

Department of Electrical Engineering
Illinois Institute of Technology
Chicago, IL 60616

October, 1983

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

A minimum distance $d_{min} = 4$ extended Reed-Solomon (RS) code over $GF(2^b)$ is constructed. The code can be used to correct any single-byte-error and simultaneously detect any double-byte-error. Fast encoding and decoding can be achieved due to some nice features of the code described in the following.

I. CODE CONSTRUCTION

Consider the RS code with generator polynomial given by

$$g(x) = (x+1)(x+\alpha)(x+\alpha^2), \tag{1}$$

where α is a primitive element of $GF(2^b)$. The code has minimum distance $d_{min} = 4$, and the parity-check matrix takes the form

$$\underline{H}_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots\cdots & 1 \\ 1 & \alpha & \alpha^2 & \alpha^3 & \cdots\cdots & \alpha^{n_1-1} \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \cdots\cdots & \alpha^{2n_1-2} \end{bmatrix}, \tag{2}$$

where $n_1 = 2^b-1$. The matrix $\underline{H}_1$ is modified by adding the identity matrix $\underline{I}_{3\times 3}$ on the left. This forms a new matrix $\underline{H}$

$$\underline{H} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 1 & \cdots\cdots & 1 \\ 0 & 1 & 0 & 1 & \alpha & \alpha^2 & \alpha^3 & \cdots\cdots & \alpha^{n_1-1} \\ 0 & 0 & 1 & 1 & \alpha^2 & \alpha^4 & \alpha^6 & \cdots\cdots & \alpha^{2n_1-2} \end{bmatrix}$$

$$= [\ \underline{I}_{3\times 3} \mid \underline{H}_1\ ]\ . \tag{3}$$

This is a $3\times n(n = n_1+3 = 2^b+2)$ matrix. Now we show that the above $\underline{H}$ matrix is a parity-check matrix for an (n, n_1) extended RS code with minimum distance $d_{min} = 4$!

The following theorem regarding the $\underline{H}$ matrix of a binary block code still holds true in the case of a nonbinary code [1]. We repeat it here.

Theorem: A code defined by a parity-check matrix $\underline{H}$ will correct single-byte-errors and simultaneously detect any combination of two byte-errors if and only if every combination of three or fewer columns of $\underline{H}$ is linearly independent.

Consider the $\underline{H}$ matrix in (3). It is obvious that

1) $\underline{H}$ contains no zero columns,

2) No two columns of $\underline{H}$ are linearly dependent.

Now we show that

3) No three columns of $\underline{H}$ are linearly dependent.

First note that every combination of three columns of $\underline{H}_1$ are linearly independent. Then for $i \neq j$ we have

i)
$$\det\begin{vmatrix} 1 & 1 & 1 \\ 0 & \alpha^i & \alpha^j \\ 0 & \alpha^{2i} & \alpha^{2j} \end{vmatrix} = \det\begin{vmatrix} \alpha^i & \alpha^j \\ \alpha^{2i} & \alpha^{2j} \end{vmatrix} = \alpha^{i+j}(\alpha^i + \alpha^j).$$

Because α is assumed to be primitive, $\alpha^i + \alpha^j \neq 0$ for $i \neq j$. Therefore

$$\det\begin{vmatrix} 1 & 1 & 1 \\ 0 & \alpha^i & \alpha^j \\ 0 & \alpha^{2i} & \alpha^{2j} \end{vmatrix} \neq 0.$$

Similarly,

$$\det\begin{vmatrix} 0 & 1 & 1 \\ 1 & \alpha^i & \alpha^j \\ 0 & \alpha^{2i} & \alpha^{2j} \end{vmatrix} = \alpha^{2i} + \alpha^{2j} = (\alpha^i + \alpha^j)^2 \neq 0$$

and

$$\det\begin{vmatrix} 0 & 1 & 1 \\ 0 & \alpha^i & \alpha^j \\ 1 & \alpha^{2i} & \alpha^{2j} \end{vmatrix} = \alpha^i + \alpha^j \neq 0.$$

ii)

$$\det\begin{vmatrix} 1 & 0 & 1 \\ 0 & 1 & \alpha^i \\ 0 & 0 & \alpha^{2i} \end{vmatrix} = \alpha^{2i} \neq 0.$$

$$\det\begin{vmatrix} 1 & 0 & 1 \\ 0 & 0 & \alpha^i \\ 0 & 1 & \alpha^{2i} \end{vmatrix} = \alpha^{i} \neq 0.$$

$$\det\begin{vmatrix} 0 & 0 & 1 \\ 1 & 0 & \alpha^i \\ 0 & 1 & \alpha^{2i} \end{vmatrix} = 1 \neq 0.$$

Therefore no three columns of $\underline{H}$ are linearly dependent.

4) Not all combinations of four columns in $\underline{H}$ are linear independent. For example,

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \alpha^i \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + \alpha^{2i} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ \alpha^i \\ \alpha^{2i} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

From 1), 2), 3), and 4) we conclude that the extended $(n, n_1) = (n = 2^b+2, n_1 = 2^b-1)$ RS code defined by the parity-check matrix in (3) has $d_{min} = 4$.

From (3) we see that the $\underline{H}$ matrix satisfies the following important considerations for an optimum code that can be used for correcting single-byte-errors and detecting double-byte-errors.

1) $\underline{H}$ is in systematic form, hence $\underline{G}$ - the generat · matrix is also in the systematic form:

$$\underline{G} = [\, \underline{H}_1^T \mid \underline{I} \,]$$

This suggests 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 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 (3), 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.

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

II. ERROR CORRECTION AND ERROR DETECTION.

The code described above has $d_{min} = 4$. Therefore it can correct single-byte-errors and simultaneously detect any double-byte-error.

1) Single byte error correction

Suppose a single error of value e occurs at byte position i. Then the syndrome is given by

$$\underline{s}_i = e\underline{h}_i = \begin{bmatrix} s_0 \\ s_1 \\ s_2 \end{bmatrix}, \tag{4}$$

where $\underline{h}_i$ is the i-th column of $\underline{H}$, $0 \leq i \leq n-1$. Note that the first nonzero element of every column of $\underline{H}$ is a unit element α^0, and $e\,\alpha^0 = e$. Therefore the error value e is given directly by the first nonzero element of the syndrome. The location of the error byte is reduced to finding a column $\underline{h}_i$ of $\underline{H}$ which satisfies the identity

$$e\underline{h}_i = \underline{s}_i. \tag{5}$$

This can be done in the following way.

Check the elements of the syndrome $\underline{s}_i$ 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, from

$$e\underline{h}_i = e\begin{bmatrix} 1 \\ \alpha^{(i-3)} \\ \alpha^{2(i-3)} \end{bmatrix} = \begin{bmatrix} s_0 \\ s_1 \\ s_2 \end{bmatrix},$$

we have

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

and i gives the error byte location, $3 \leq i \leq n-1$.

2) Double-byte-error detection

Because the code is double-byte-error detecting, the sum of any two syndromes corresponding to two single-byte-errors e_i and e_j $(i \neq j)$ is not equal to any single-byte-error syndrome $\underline{s}_k$, that is,

$$\underline{s}_i + \underline{s}_j \neq \underline{s}_k \quad \text{for} \quad i \neq j.$$

Using this property, double-byte-error detection can be done in the following way. If

$$s_{i_1} = 0,\ s_{i_2} \neq 0,\ s_{i_3} \neq 0, \quad \text{where} \quad i_1, i_2, i_3 \quad \varepsilon (0, 1, 2),$$

or if

$$s_0 \neq 0,\ s_1 \neq 0,\ s_2 \neq 0 \quad \text{and} \quad \frac{s_1}{s_0} \neq \frac{s_2}{s_1}$$

then a double-byte-error is detected.

REFERENCES

1. S. Lin and D.J. Costello, Jr., Error Control Coding: Fundamentals and Applications, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1983.