[3] H. A. Spang, III, "Optimum control of an unknown linear plant using Bayesian estimation of the error," G.E. Research Lab., Schenectady, N. Y., Rept. 64-RL-3703E, July 1964.
[4] D. V. Lindley, "The use of prior probability distributions in statistical inference and decisions," *1960 Proc. 4th Berkeley Symp. on Mathematical Statistics and Probability*, vol. 1, pp. 453–468.
[5] D. Blackwell and M. A. Girshick, *Theory of Games and Statistical Decisions*. New York: Wiley, 1954.
[6] E. B. Dynkin, "Necessary and sufficient statistics for a family of probability distributions," *Selected Translations in Mathematical Statistics and Probability*, vol. 1, 1961, pp. 17–40.
[7] L. D. Brown, "Sufficient statistics in the case of independent random variables," in *The Annals of Mathematical Statistics*, vol. 35, 1964, pp. 1456–1474.
[6] R. Bellman, *Adaptive Control Processes, a Guided Tour*. Princeton, N. J.: Princeton Univ. Press, 1961, p. 215.
[9] N. Abramson and D. Braverman, "Learning to recognize patterns in a random environment," *IRE Trans. on Information Theory*, vol. IT-8, pp. 58–63, September 1962.
[10] D. G. Keehn, "A note on learning for gaussian properties," *IEEE Trans. on Information Theory*, vol. IT-11, pp. 126–132, January 1965.
[11] G. Turin, "Communication through noisy, random-multipath channels," M.I.T. Lincoln Lab., Lexington, Mass., TR No. 116, May 1956.
[12] H. Raiffa and R. Schlaifer, "Applied statistical decision theory," Div. of Research, Harvard Business School, Harvard Univ., Boston, Mass., 1961.
[13] R. F. Daly, "The adaptive binary-detection problem on the real line," Stanford Electronics Labs., Stanford, Calif., TR No. 2003-3, February 1962.

On Decoding BCH Codes

G. DAVID FORNEY, JR., MEMBER, IEEE

Abstract—**The Gorenstein-Zierler decoding algorithm for BCH codes is extended, modified, and analyzed; in particular, we show how to correct erasures as well as errors, exhibit improved procedures for finding error and erasure values, and consider in some detail the implementation of these procedures in a special-purpose computer.**

I. Introduction

THE DISCOVERY of the binary codes of Bose and Ray-Chaudhuri [1], [2] and (independently) Hocquenghem [3] has been, perhaps, the outstanding success of the search for codes based on algebraic structures. Not the least of their virtues is their capability of being decoded by relatively straightforward algorithms. Peterson [4] was the first to outline an efficient decoding procedure, which was actually realized by Bartee and Schneider [5] in a small special-purpose computer.

Recent work has focused attention on the multisymbol generalizations of these codes. These were first considered by Gorenstein and Zierler [6], who developed an error-correcting algorithm for them.

This paper reports extensions, modifications, and analyses of these decoding procedures. We shall be concerned with the general codes, which we shall call BCH codes, but all of our results apply to the binary special case. In particular, we shall proceed as follows:

1) to extend the GZ algorithm to correct erasures, as well as errors, by introducing a modified set of parity checks;

2) to improve the final step of the GZ algorithm by giving explicit formulas for error values, thereby eliminating the need for solving simultaneous equations;

3) to introduce an alternative method for determining error values, which has a use when the number of errors to be corrected is small;

4) to note a method for solving for erasures separately from errors;

5) to exhibit a generalization of Chien's [7] 'direct method' of locating errors; and

6) to analyze in some detail the implementation of these procedures in a computer with finite-field arithmetic unit, and determine that the number of operations increases only as a small power of the number of errors to be corrected.

Also, for the reader who likes to confirm his understanding by working out simple examples, we have included a decoding problem, with the aid of which the different procedures can be illustrated.

II. Preliminary Definitions

BCH codes are conveniently described in the language of the theory of finite fields, which has been well developed for this purpose by Peterson [8]. A finite, or Galois, field with p^M elements [written $GF(p^M)$] exists if p is a prime, and M is any integer; p is called the characteristic of the field. In any field there is a zero element 0, a unit element 1, and at least one primitive element α, such that any other nonzero element β can be expressed as a power of α. The order of β is the least integer e such that $\beta^e = 1$; a primitive element α has order $p^M - 1$. If M is a factor of N, the elements of $GF(p^M)$ are included in $GF(p^N)$, and

Manuscript received January 31, 1965; revised July 1, 1965. The work reported in this paper was supported in part by the Joint Services Electronics Program under Contract DA36-039-AMC-03200(E); and in part by the National Science Foundation (Grant GP-2495), the National Institute of Health (Grant MH-04737-04) and the National Aeronautics and Space Administration (Grants NsG 334 and NsG 496). Portions of this work have appeared in Quarterly Progress Report No. 76, M.I.T. Research Laboratory of Electronics, Cambridge, Mass., p. 236, January 1965.

The author is with the Codex Corporation, Watertown, Mass. He was formerly with the M.I.T. Research Laboratory of Electronics, Cambridge, Mass.

the former is said to be a subfield of the latter, or the latter an extension field of the former.

For example, $GF(2^4)$ consists of the elements 0, 1, α, α^2, $\cdots$ α^{14}; 0 and 1 are the subfield $GF(2)$, while 0, 1, α^5, and α^{10} are the subfield $GF(2^2)$. A particular representation for $GF(2^4)$, based on polynomials modulo the irreducible (over $GF(2)$) polynomial $X^4 + X + 1$, is given in Peterson [8] as

0	0000	α^3	0001	α^7	1101	α^{11}	0111
1	1000	α^4	1100	α^8	1010	α^{12}	1111
α	0100	α^5	0110	α^9	0101	α^{13}	1011
α^2	0010	α^6	0011	α^{10}	1110	α^{14}	1001

Since $\alpha^{15} = 1$, $\alpha^i \cdot \alpha^j = \alpha^{(i+j) \bmod 15}$; addition of two elements in the field is achieved by bit-by-bit modulo 2 addition of the two representations.

In this language, code words of length n_0 are represented by sequences of n_0 elements from $GF(p^M)$, which we shall write as a vector

$$\mathbf{f} \equiv (f_1, f_2, \cdots, f_{n_0}).$$

If we define $\mathbf{X}_{(a,b)}$ as the column vector of descending powers of X,

$$\mathbf{X}_{(a,b)} \equiv (X^a, X^{a-1}, \cdots, X^b)^T,$$

where X is an indeterminate and T indicates the transpose, then the dot product

$$\mathbf{f} \cdot \mathbf{X}_{(n_0-1,0)} = \sum_i f_i X^{n_0-i} \equiv f(X)$$

is a polynomial in X of degree $n_0 - 1$, which we define as $f(X)$. Similarly, if β is any element of $GF(p^N)$ or of an extension or subfield thereof, we can define

$$f(\beta^m) \equiv \mathbf{f} \cdot \mathbf{\beta}^{\mathbf{m}}_{(n_0-1,0)} = \sum_i f_i \beta^{m(n_0-i)},$$

where

$$\mathbf{\beta}^{\mathbf{m}}_{(a,b)} \equiv (\beta^{ma}, \beta^{m(a-1)}, \cdots, \beta^{mb})^T.$$

If the order of β is n_0, BCH codes consist of the set of all $\mathbf{f}$ such that $f(\beta^m) = 0$, for all m in the range $m_0 \leq m \leq m_0 + d - 2$, where m_0 and d are arbitrary integers. It develops [8] that d is a lower bound to the minimum distance of the code. Information on the number of words in some binary BCH codes is given in Peterson [8]. Commonly, β is taken as a primitive element of $GF(p^N)$, and m_0 equals 0 or 1.

In examples, we shall use a Reed-Solomon [8], [9] code of length 15, with elements from $GF(2^4)$, such that $\mathbf{f}$ is in the code if $f(\alpha) = f(\alpha^2) = \cdots = f(\alpha^8) = 0$, where α is a primitive element of $GF(2^4)$. Such a code has $(2^4)^7$ words, or 7 information symbols, and its minimum distance is 9. In our examples with this code, $GF(2^4)$ will be represented as previously.

In an actual communications system, a modulator can transmit one of p^M signals corresponding to the p^M field elements; a code word is physically realized by a sequence of n_0 such signals. At the receiver, the function of the demodulator is to guess which of the p^M signals was sent; n_0 of these guesses constitute a received word, and can be represented by a vector

$$\mathbf{r} \equiv (r_1, r_2, \cdots, r_{n_0})$$

of n_0 elements from $GF(p^M)$. If the ith guess is correct, $r_i = f_i$; if it is incorrect, we say an error has been made; the value of the error is defined as $r_i - f_i$ and the locator of the error as β^{n_0-i}. If there are t errors in all, we shall denote the values by e_j and the locators by X_j, $1 \leq j \leq t$.

From the received word one obtains the parity checks S_m defined by

$$S_m \equiv r(\beta^m) = \sum_{i=1}^{n_0} r_i \beta^{m(n_0-i)} = \mathbf{r} \cdot \mathbf{\beta}^{\mathbf{m}}_{(n_0-1,0)},$$

$$m_0 \leq m \leq m_0 + d - 2$$

From the definitions of e_j and X_j,

$$S_m = f(\beta^m) + \sum_{j=1}^{t} e_j X_j^m;$$

but $f(\beta^m) = 0$ for $m_0 \leq m \leq m_0 + d - 2$ and for all code words $\mathbf{f}$, so that

$$S_m = \sum_{j=1}^{t} e_j X_j^m, \qquad m_0 \leq m \leq m_0 + d - 2. \tag{1}$$

The decoding problem is to solve (1), called the parity check equations, for the e_j and X_j. Whenever $2t < d$, the algorithm of Gorenstein and Zierler [6] solves this problem. The algorithm consists of three steps: first the number is found, then the locations, and finally the value of the errors, as we shall explain in more detail later.

In addition to the column vectors $\mathbf{X}_{(a,b)}$ and $\mathbf{\beta}^{\mathbf{m}}_{(a,b)}$ already defined, we shall use the column vectors $\mathbf{X}_{j(a,b)}$, $\mathbf{Y}_{k(a,b)}$, and $\mathbf{Z}_{(a,b)}$ of descending powers of X_j, Y_k, and Z, respectively, in which Y_k and Z have yet to be introduced. We shall also use the column vectors

$$\mathbf{S}_{(a,b)} \equiv (S_a, S_{a-1}, \cdots, S_b)^T,$$

$$m_0 \leq b \leq a \leq m_0 + d - 2$$

and

$$\mathbf{T}_{(a,b)} \equiv (T_a, T_{a-1}, \cdots, T_b)^T, \qquad 0 \leq b \leq a \leq d - s - 2,$$

where the S_m are the parity checks introduced above, and the T_n are the modified cyclic parity checks to be introduced below. With these expressions and (1) we have, for example,

$$\mathbf{S}_{(a,b)} = \sum_{j=1}^{t} e_j \mathbf{X}_{j(a,b)}.$$

Finally, let us consider the polynomial $\sigma(Z)$ defined by

$$\sigma(Z) \equiv (Z - Z_1)(Z - Z_2) \cdots (Z - Z_L),$$

where Z is an indeterminate, and the Z_l are members of a field. Clearly, $\sigma(Z) = 0$ if and only if Z equals one of the Z_l. Expanding $\sigma(Z)$, we get

$$\sigma(Z) = Z^L - (Z_1 + Z_2 + \cdots + Z_L)Z^{L-1} + \cdots + (-1)^L(Z_1Z_2 \cdots Z_L).$$

The coefficient of $(-1)^{L-i}Z^i$ in this expansion is defined as the $(L - l)$th elementary symmetric function σ_{L-l} of the Z_i; note that σ_0 is always one. We define $\mathbf{\sigma}$ as the row vector

$$(\sigma_0, -\sigma_1, \cdots, (-1)^L\sigma_L);$$

then the dot product

$$\mathbf{\sigma}\cdot\mathbf{Z}_{(L,0)} = \sigma(Z).$$

III. Decoding Algorithm for Erasures and Errors

Coding is required in communication systems to combat the errors that occur in the guesses of the demodulator, as we have seen. It has long been recognized [10] that there are advantages in allowing the demodulator not to guess at all on certain transmissions when the evidence does not clearly indicate one signal as the most probable; such events are called erasures. It is convenient to imagine that in the event of an erasure the demodulator does make some guess, perhaps arbitrary, but in addition passes on the side information to the decoder that this guess is absolutely unreliable and is to be disregarded.

From information theoretic considerations, allowing the option of erasures is thus a way of passing more information to the decoder about the signal actually received. In the context of the BCH algorithm, an erasure can be regarded as an error whose location is known and therefore does not have to be computed; the use of erasures should allow the shifting of some of the burden of determining error locations from the decoder to the demodulator.

As with errors, if an erasure occurs in the ith place, we say its value is $r_i - f_i$, where r_i represents the arbitrary guess of the demodulator; the value of an erasure may be zero, while that of an error may not. β^{n_0-i} is the locator of the erasure, which is known to the decoder, while that of an error is not. If there are s erasures, we denote their values by d_k and their locators by Y_k, $1 \leq k \leq s$. The values and locators of the t errors which may also occur will continue to be denoted by e_j and X_j, $1 \leq j \leq t$.

We observe that in this case (1), the parity check equations, become

$$S_m = \sum_{j=1}^{t} e_jX_j^m + \sum_{k=1}^{s} d_kY_k^m, \quad m_0 \leq m \leq m_0 + d - 2. \tag{2}$$

In our vectorial notation we have

$$\mathbf{S}_{(a,b)} = \sum_{j=1}^{t} e_j\mathbf{X}_{j(a,b)} + \sum_{k=1}^{s} d_k\mathbf{Y}_{k(a,b)},$$

$$m_0 \leq b \leq a \leq m_0 + d - 2. \tag{2a}$$

The decoding problem is now to find the e_j, X_j, and d_k, given the S_m and Y_k. The modification of the GZ algorithm now to be presented permits solution of this problem whenever $2t + s < d$. Essentially we derive from the $d - 1$ parity check equations of (2) a set of $d - s - 1$ equations of the same form as (1), to which therefore the GZ algorithm can be applied.

Define

$$\sigma_d(Z) \equiv (Z - Y_1)(Z - Y_2) \cdots (Z - Y_s)$$

and let $\mathbf{\sigma}_d$ then be the vector of the symmetric functions σ_{dk} of the erasure locators Y_k, as above. We define the modified cyclic parity checks T_n by

$$T_n \equiv \mathbf{\sigma}_d\cdot\mathbf{S}_{(m_0+n+s,m_0+n)}, \qquad 0 \leq n \leq d - s - 2. \tag{3}$$

The range of n is restricted to $0 \leq n \leq d - s - 2$, since we must have $m_0 \leq m_0 + n$ and $m_0 + n + s \leq m_0 + d - 2$. (In the case of no erasures, $T_n = S_{m_0+n}$.)

Combining (2a) and (3), we have

$$\begin{aligned} T_n &\equiv \sum_{j=1}^{t} e_j\mathbf{\sigma}_d\cdot\mathbf{X}_{j(m_0+n+s,m_0+n)} + \sum_{k=1}^{s} d_k\mathbf{\sigma}_d\cdot\mathbf{Y}_{k(m_0+n+s,m_0+n)} \\ &= \sum_{j=1}^{t} e_jX_j^{m_0}X_j^n\sigma_d(X_j) + \sum_{k=1}^{s} d_kY_k^{m_0+n}\sigma_d(Y_k) \qquad (4) \\ &= \sum_{j=1}^{t} E_jX_j^n, \qquad 0 \leq n \leq d - s - 2. \end{aligned}$$

Here we have defined

$$E_j = e_jX_j^{m_0}\sigma_d(X_j)$$

and used the fact that $\sigma_d(Y_k) = 0$, since Y_k is one of the erasure locators upon which $\mathbf{\sigma}_d$ is defined.

In (4) we now have $d - s - 1$ equations of exactly the form of (1), which are thus soluble for the E_j and X_j by the GZ algorithm whenever $2t < d - s$. In particular, we have the following lemma and theorem whose proofs, being identical to those which appear in Peterson [8], are omitted. We define t_0, the maximum correctable number of errors, as the greatest integer for which $2t_0 < d - s$. Then we have

Lemma: If $t \leq t_0$, then the $t \times t$ matrix M_t has rank t, where

$$M_t \equiv \begin{bmatrix} T_{2t_0-2} & T_{2t_0-3} & \cdots & T_{2t_0-t-1} \\ T_{2t_0-3} & T_{2t_0-4} & \cdots & T_{2t_0-t-2} \\ \vdots & \vdots & & \vdots \\ T_{2t_0-t-1} & T_{2t_0-t-2} & \cdots & T_{2t_0-2t} \end{bmatrix}.$$

Theorem: If $t \leq t_0$, then the $t_0 \times t_0$ matrix M has rank t, where

$$M \equiv \begin{bmatrix} T_{2t_0-2} & T_{2t_0-3} & \cdots & T_{t_0-1} \\ T_{2t_0-3} & T_{2t_0-4} & \cdots & T_{t_0-2} \\ \vdots & \vdots & & \vdots \\ T_{t_0-1} & T_{t_0-2} & \cdots & T_0 \end{bmatrix}.$$

This theorem allows determination of t from the T_n, which is the first step in the algorithm.

Now consider the vector $\mathbf{\sigma}_e$ of elementary symmetric functions σ_{ej} of the X_j, and its associated polynomial

$$\sigma_e(X) = \mathbf{\sigma}_e\cdot\mathbf{X}_{(t,0)}.$$

We have from (4) and the fact that $\sigma_e(X_j) = 0, 1 \leq j \leq t$,

$$\mathbf{o}_e \cdot \mathbf{T}_{(n'+t,n')} = \sum_{j=1}^{t} E_j X_j^{n'} \sigma_e(X_j) = 0,$$

$$0 \leq n' \leq d - s - t - 2.$$

σ_{e0} always equals one; therefore this gives us $d - s - t - 1$ linear equations in t unknowns. Since $2t + s < d$, $t \leq d - s - t - 1$; thus we have sufficient equations to solve for the σ_{ej}, $1 \leq j \leq t$. By defining the vector

$$\mathbf{o}'_e \equiv (-\sigma_{e1}, \sigma_{e2}, \cdots, (-1)^t \sigma_{et})$$

the equations specified by $2t_0 - 2t \leq n' \leq 2t_0 - t - 1$ can be expressed in matrix form as

$$-\mathbf{T}_{(2t_0-1,2t_0-t)} = \mathbf{o}'_e M_t, \tag{5}$$

where M_t is as in the lemma and therefore invertible; thus the equations are soluble for $\mathbf{o}'_e$ and hence $\mathbf{o}_e$. Then since $\sigma_e(\beta^{n_0-i})$ is zero if and only if β^{n_0-i} is an error locator, calculation of $\sigma_e(\beta^{n_0-i})$ for each i will reveal in turn the locations of all t errors, which concludes the second step in the algorithm.

A. Remarks I

In Peterson [8], first the rank of M is found, and then a set of t equations in t unknowns is solved, as in the present work. We remark that with the definition of M_t in the foregoing lemma, these two steps may be combined into one (as is implicit in Gorenstein and Zierler). Consider the equation

$$-\mathbf{T}_{(2t_0-1,t_0)} = \mathbf{o}''_e M, \tag{6}$$

where $\mathbf{o}''_e \equiv (-\sigma_{e1}, \sigma_{e2}, \cdots, (-1)^t \sigma_{et}, 0, \cdots, 0)$. An efficient way of solving (6) is by a Gauss-Jordan reduction to upper triangular form. Since the rank of M is t, this reduction will leave t nontrivial equations, the last $t_0 - t$ equations being simply $0 = 0$. But now M_t is the upper left-hand corner of M, so that the upper left-hand corner of the reduced M will be the reduced M_t. We can therefore set the last $t_0 - t$ components of $\mathbf{o}''_e$ to zero, and get a set of equations equivalent to (5), which can be solved for $\mathbf{o}'_e$. Thus we need only one reduction, not two; since Gauss-Jordan reductions are tedious, this may be a significant saving.

This procedure works whenever $t \leq t_0$; that is, whenever the received word lies within distance t_0 of some code word, not counting places in which there are erasures. It will generally be possible to receive words greater than distance t_0 from any code word, and upon such words this procedure must fail. This failure, corresponding to a detectable error, must turn up either in the failure of (6) to be reducible to the form previously described, or in $\sigma_e(X)$ having an insufficient number of nonzero roots of the form β^{n_0-i}; either of these events may occur [11].

Finally, if $d - s$ is even, the preceding algorithm will locate all errors when $t \leq t_0 = (d - s - 2)/2$. Also, if $t = t_0 + 1$, an uncorrectable error can be detected by the nonvanishing of the determinant of the $t \times t$ matrix with T_{d-s-2} in the upper left, T_0 in the lower right. Such an error would be detected at some later stage in the correction process, however.

B. Example 1

Consider the (15, 7), distance 9 Reed-Solomon code introduced earlier. Suppose there occur errors of value α^4 in the first position and α in the fourth position, and erasures of value 1 in the second position and α^7 in the third position.

$$(e_1 = \alpha^4, X_1 = \alpha^{14}, e_2 = \alpha, X_2 = \alpha^{11},$$
$$d_1 = 1, Y_1 = \alpha^{13}, d_2 = \alpha^7, Y_2 = \alpha^{12}).$$

In this case the parity checks S_m will turn out to be

$$S_1 = \alpha^{14}, \quad S_2 = \alpha^{13}, \quad S_3 = \alpha^5, \quad S_4 = \alpha^6,$$
$$S_5 = \alpha^9, \quad S_6 = \alpha^{13}, \quad S_7 = \alpha^{10}, \quad \text{and} \quad S_8 = \alpha^4.$$

With these eight parity checks and two erasure locators, the decoder must find the number and position of the errors. First it forms $\mathbf{o}_d = (\sigma_{d0}, \sigma_{d1}, \sigma_{d2})$. (Since we are working in a field of characteristic two, where addition and subtraction are identical, we omit minus signs.)

$$\sigma_{d0} = 1$$

$$\sigma_{d1} = Y_1 + Y_2 = \alpha^{13} + \alpha^{12}$$
$$= (1011) + (1111) = (0100) = \alpha$$

$$\sigma_{d2} = Y_1 Y_2 = \alpha^{13} \cdot \alpha^{12} = \alpha^{10}.$$

Next it forms the six modified cyclic parity checks T_n by (3),

$$T_0 = S_3 + \sigma_{d1} S_2 + \sigma_{d2} S_1$$
$$= \alpha^5 + \alpha \cdot \alpha^{13} + \alpha^{10} \cdot \alpha^{14} = \alpha^5 + \alpha^{14} + \alpha^9$$
$$= (0110) + (1001) + (0101) = (1010) = \alpha^8$$

$$T_1 = S_4 + \sigma_{d1} S_3 + \sigma_{d2} S_2 = \alpha^8$$

$$T_2 = 0, T_3 = \alpha^3, T_4 = \alpha^{13}, T_5 = \alpha^3.$$

Equation (6) now takes the form

$$\begin{aligned} \alpha^3 &= \alpha^{13}\sigma_{e1} + \alpha^3 \sigma_{e2} \\ \alpha^{13} &= \alpha^3 \sigma_{e1} + \alpha^8 \sigma_{e3} \\ \alpha^3 &= \alpha^8 \sigma_{e2} + \alpha^8 \sigma_{e3}. \end{aligned}$$

With these equations reduced to upper triangular form, the decoder gets

$$\begin{aligned} \alpha^5 &= \sigma_{e1} + \alpha^5 \sigma_{e2} \\ \alpha^{10} &= \sigma_{e2} + \sigma_{e3} \\ 0 &= 0. \end{aligned}$$

From the vanishing of the third equation, it learns that only two errors actually occured. Therefore it sets σ_{e3} to zero and solves for σ_{e1} and σ_{e2}, obtaining $\sigma_{e2} = \alpha^{10}$, $\sigma_{e1} = \alpha^{10}$. Finally, it evaluates the polynomial

$$\sigma_e(X) = X^2 + \sigma_{e1} X + \sigma_{e2} = X^2 + \alpha^{10} X + \alpha^{10}$$

for X equal to each of the nonzero elements of $GF(2^4)$; $\sigma_e(X) = 0$ when $X = \alpha^{14}$ and $X = \alpha^{11}$, so that these are the two error locators.

IV. Solving for the Values of the Erased Symbols

Once the errors have been located, they can be treated as erasures. The third and final step of the algorithm is then to determine the values of $s + t$ erased symbols, given that there are no errors in the remaining symbols. To simplify notation, we consider the problem of finding the d_k, given Y_k, $1 \leq k \leq s$, and $t = 0$.

Equation (2) is a set of linear equations in the erasure values, which can be solved by standard techniques. Its particular form gives us another approach, however, which is more efficient.

The derivation of (7), which follows, can be understood by imagining the following. Suppose we wanted to find d_{k_0}. If we continued to treat the remaining $s - 1$ erasures as erasures, but made a stab at guessing d_{k_0}, we would get a word with $s - 1$ erasures and either one or (on the chance of a correct guess) zero errors. The rank of the matrix M_1 would therefore be either zero or one; but M_1 is simply a single modified cyclic parity check, formed from the elementary symmetric functions of the $s - 1$ remaining erasure locators. Its vanishing would therefore tell us when we had guessed d_{k_0} correctly.

Symbolically, let ${}_{k_0}\boldsymbol{\delta}_d$ be the vector of elementary symmetric functions of the $s - 1$ erasure locators not including Y_{k_0}.

Since $t = 0$, we have from (2a)

$$\mathbf{S}_{(m_0+d-2,m_0+d-s-1)} = \sum_{k=1}^{s} d_k Y_k^{m_0+d-s-1} \mathbf{Y}_{k(s-1,0)}$$

and therefore

$$\begin{aligned} {}_{k_0}T_{d-s-1} &\equiv {}_{k_0}\boldsymbol{\delta}_d \cdot \mathbf{S}_{(m_0+d-2,m_0+d-s-1)} \\ &= d_{k_0} Y_{k_0}^{m_0+d-s-1} {}_{k_0}\sigma_d(Y_{k_0}) + \sum_{k \neq k_0} d_k Y_k^{m_0+d-s-1} {}_{k_0}\sigma_d(Y_k) \\ &= d_{k_0} Y_{k_0}^{m_0+d-s-1} {}_{k_0}\sigma_d(Y_{k_0}), \end{aligned}$$

since ${}_{k_0}\sigma_d(Y_k) = 0$, $k \neq k_0$. Thus

$$d_{k_0} = \frac{{}_{k_0}T_{d-s-1}}{Y_{k_0}^{m_0+d-s-1} {}_{k_0}\sigma_d(Y_{k_0})}.$$

This yields an explicit formula for d_{k_0} which is valid for any s

$$d_{k_0} = \frac{S_{m_0+d-2} - {}_{k_0}\sigma_{d1} S_{m_0+d-3} + {}_{k_0}\sigma_{d2} S_{m_0+d-4} - \cdots}{Y_{k_0}^{m_0+d-2} - {}_{k_0}\sigma_{d1} Y_{k_0}^{m_0+d-3} + {}_{k_0}\sigma_{d2} Y_{k_0}^{m_0+d-4} - \cdots}. \quad (7)$$

Evidently we can find all erasure values in this way; each requires the calculation of the symmetric functions of a different set of $s - 1$ locators. Alternatively, after finding d_1, we could modify all parity checks to account for this information $[\mathbf{S}'_{(m_0+d-2,m_0)} = \mathbf{S}_{(m_0+d-2,m_0)} - d_1 \mathbf{Y}_{1(m_0+d-2,m_0)}]$, and solve for d_2 in terms of these new parity checks and the remaining $s - 2$ erasure locators, and so forth.

A. Example 2

As a continuation of our previous example, let the decoder solve for e_1. The elementary symmetric functions of X_2, Y_1, and Y_2 are

$$\begin{aligned} \sigma_3 &= X_2 Y_1 Y_2 = \alpha^6, \\ \sigma_2 &= Y_2 Y_1 + X_2 Y_2 + X_2 Y_1 = \alpha^3, \\ \sigma_1 &= X_2 + Y_1 + Y_2 = \alpha^6. \end{aligned}$$

Therefore

$$e_1 = \frac{\alpha^4 + \alpha^6 \cdot \alpha^{10} + \alpha^3 \cdot \alpha^{13} + \alpha^6 \cdot \alpha^9}{\alpha^7 + \alpha^6 \cdot \alpha^8 + \alpha^3 \cdot \alpha^9 + \alpha^6 \cdot \alpha^{10}} = \frac{\alpha}{\alpha^{12}} = \alpha^4.$$

In a similar manner, e_2 can be found, or the decoder can calculate

$$\begin{aligned} S'_8 &= S_8 + \alpha^4 X_1^8 = \alpha^{13}, \\ S'_7 &= S_7 + \alpha^4 X_1^7 = \alpha^3, \\ S'_6 &= S_6 + \alpha^4 X_1^6 = 0. \end{aligned}$$

Since

$$\sigma'_2 = Y_1 Y_2 = \alpha^{10}, \qquad \sigma'_1 = Y_1 + Y_2 = \alpha,$$

$$e_2 = \frac{\alpha^{13} + \alpha \cdot \alpha^3}{\alpha^{13} + \alpha \cdot \alpha^2 + \alpha^{10} \cdot \alpha^6} = \frac{\alpha^{11}}{\alpha^{10}} = \alpha.$$

Third, $S''_8 = \alpha^2$, $S''_7 = 0$,

$$d_1 = \frac{\alpha^2}{\alpha + \alpha^{12} \cdot \alpha^{13}} = 1$$

and finally, with

$$S'''_8 = \alpha^{13}, \qquad d_2 = \frac{\alpha^{13}}{\alpha^6} = \alpha^7.$$

B. Remarks II

By similar reasoning, we find

$$e_{j_0} = \frac{{}_{j_0}\boldsymbol{\delta}_e \cdot \mathbf{T}_{(d-s-2,d-s-t-1)}}{X_{j_0}^{m_0+d-s-t-1} {}_{j_0}\sigma_e(X_{j_0}) \sigma_d(X_{j_0})}$$

which gives the error values in terms of the modified cyclic parity checks. We could therefore find all error values by this formula, modify the parity checks S_m accordingly, and then solve for the erasure values by (7).

These results are also obtainable from the first s parity check equations by solving for the s unknown erasure values by Cramer's Rule, with use of explicit formulas for the determinants of van der Monde-like matrices with missing powers.

Reed-Solomon [8], [9] codes are the subclass of BCH codes for which the symbol values and locators are defined on the same field ($M = N$). The number of check symbols in an RS code is equal to $d - 1$, the maximum correctable number of erasures. One way of generating a systematic Reed-Solomon code—that is, a code in which the first k symbols are arbitrary information symbols, while the last $n_0 - k$ are check symbols—would be to use (7) to solve for the last $n_0 - k$ symbols in terms of the parity

checks S_m of the first k symbols, as though they were erasures. Since the 'erasures' are always in the last $n_0 - k$ places, each of the final $n_0 - k$ symbols can be expressed as a fixed linear function of the S_m.

V. An Alternative Determination of Error Values

The point of view which led us to the erasure correction procedure just described leads us also to another method of determining the values of the errors. Suppose the number of errors t had been discovered; then the $t \times t$ matrix M_t would have rank t and therefore nonzero determinant. If the decoder now determined the locator X_{i_0} of any error, guessed the corresponding error value e_{i_0}, and modified the T_n accordingly, then the guessed word would either still have t or (on the chance of a correct guess) $t - 1$ errors, and the $t \times t$ matrix M_t' formed from the new T_n' would have zero determinant if and only if the guess were correct.

In general, one would expect this argument to yield a polynomial in e_{i_0} of degree t as the equation of condition, but because of the special form of M_t this equation is only first degree, and an explicit formula for e_{i_0} can be obtained.

Let

$$\mathbf{S}'_{(m_0+n+s,\,m_0+n)} \equiv \mathbf{S}_{(m_0+n+s,\,m_0+n)} - e_{i_0}\mathbf{X}_{i_0(m_0+n+s,\,m_0+n)}.$$

Then

$$\begin{aligned} T_n' &\equiv \mathfrak{d}_d \cdot \mathbf{S}'_{(m_0+n+s,\,m_0+n)} \\ &= \mathfrak{d}_d \cdot \mathbf{S}_{(m_0+n+s,\,m_0+n)} - e_{i_0}\mathfrak{d}_d \cdot \mathbf{X}_{i_0(m_0+n+s,\,m_0+n)} \\ &= T_n - e_{i_0}X_{i_0}^{m_0+n}\sigma_d(X_{i_0}) = T_n - E_{i_0}X_{i_0}^{n} \end{aligned}$$

$$M_t' = \begin{bmatrix} T_{2t_0-2} - E_{i_0}X_{i_0}^{2t_0-2} & T_{2t_0-3} - E_{i_0}X_{i_0}^{2t_0-3} & \cdots & T_{2t_0-t-1} - E_{i_0}X_{i_0}^{2t_0-t-1} \\ T_{2t_0-3} - E_{i_0}X_{i_0}^{2t_0-3} & T_{2t_0-4} - E_{i_0}X_{i_0}^{2t_0-4} & \cdots & T_{2t_0-t-2} - E_{i_0}X_{i_0}^{2t_0-t-2} \\ \vdots & \vdots & & \vdots \\ T_{2t_0-t-1} - E_{i_0}X_{i_0}^{2t_0-t-1} & T_{2t_0-t-2} - E_{i_0}X_{i_0}^{2t_0-t-2} & \cdots & T_{2t_0-2t} - E_{i_0}X_{i_0}^{2t_0-2t} \end{bmatrix}$$

Let us expand this determinant into 2^t determinants, using the fact that the determinant of the matrix which has the vector $(\mathbf{a} + \mathbf{b})$ as a row is the sum of the determinants of the two matrices which have $\mathbf{a}$ and $\mathbf{b}$ in that row, respectively. We classify the resulting determinants by the number of rows which have E_{i_0} as a factor.

There is one determinant with no row containing E_{i_0}, which is simply $|M_t|$.

There are t determinants with one row having E_{i_0} as a factor. For example, the first is

$$\begin{vmatrix} -E_{i_0}X_{i_0}^{2t_0-2} & -E_{i_0}X_{i_0}^{2t_0-3} & \cdots & -E_{i_0}X_{i_0}^{2t_0-t-1} \\ T_{2t_0-3} & T_{2t_0-4} & \cdots & T_{2t_0-t-2} \\ \vdots & \vdots & & \vdots \\ T_{2t_0-t-1} & T_{2t_0-t-2} & \cdots & T_{2t_0-2t} \end{vmatrix}.$$

There are $\binom{t}{2}$ determinants with two rows having E_{i_0} as a factor. The first is

$$\begin{vmatrix} -E_{i_0}X_{i_0}^{2t_0-2} & -E_{i_0}X_{i_0}^{2t_0-3} & \cdots & -E_{i_0}X_{i_0}^{2t_0-t-1} \\ -E_{i_0}X_{i_0}^{2t_0-3} & -E_{i_0}X_{i_0}^{2t_0-4} & \cdots & -E_{i_0}X_{i_0}^{2t_0-t-2} \\ T_{2t_0-4} & T_{2t_0-5} & \cdots & T_{2t_0-t-3} \\ \vdots & \vdots & & \vdots \\ T_{2t_0-t-1} & T_{t_0-t-2} & \cdots & T_{2t_0-2t} \end{vmatrix}.$$

But in this determinant the first row is simply X_{i_0} times the second, so that the determinant is zero. Furthermore, in all such determinants with two or more rows having E_{i_0} as a factor, these rows will be some power of X_{i_0} times each other, so that all such determinants are zero.

The t determinants with one row having E_{i_0} as a factor are all linear in E_{i_0}, and contain explicit powers of X_{i_0} between $2t_0 - 2t$ and $2t_0 - 2$; their sum is then

$$-E_{i_0}X_{i_0}^{2t_0-2t}P(X_{i_0})$$

where $P(X_{i_0})$ is a polynomial of degree $2t - 2$, whose coefficients are functions of the original T_n.

Finally, we recall that $E_{i_0} = e_{i_0}X_{i_0}^{m_0}\sigma_d(X_{i_0})$ and that $|M_t'| = 0$ if and only if e_{i_0} is chosen correctly, from which we get the equation of condition

$$0 = |M_t'| = |M_t| - E_{i_0}X_{i_0}^{2t_0-2t}P(X_{i_0})$$

so

$$e_{i_0} = \frac{|M_t|}{X_{i_0}^{m_0+2t_0-2t}\sigma_d(X_{i_0})P(X_{i_0})}. \tag{8}$$

We inquire into the ease of using this formula to compute error values. $|M_t|$ can be obtained as a by-product of the reduction of (6). The only term in the denominator of (8) which is not readily calculable is $P(X_{i_0})$. In general, if A_{ik} is the determinant of the matrix remaining after the ith row and kth column are struck from M_t, then

$$P(X_{i_0}) = \sum_{l=2}^{2t} (-X_{i_0})^{2t-l} \sum_{i+k=l} A_{ik}.$$

A simplification occurs when we are in a field of characteristic two. Note that, because of the diagonal symmetry of M_t, $A_{ik} = A_{ki}$. Any sum $\sum_{i+k=l} A_{ik}$ will consist entirely of pairs $A_{ik} + A_{ki} = 0$, unless l is even, when the entire sum equals A_{ij}, with $j = l/2$. Then

$$P(X_{i_0}) = \sum_{j=1}^{t} X_{i_0}^{2(t-j)}A_{jj}. \tag{9}$$

Evaluation of the coefficients of $P(X)$ in a field of characteristic two therefore involves calculating $t \times (t-1) \times (t-1)$ determinants.

A. Example 3

Let the decoder solve (6) as before, obtaining as a by-product $|M_t| = \alpha^6$. Trivially, $A_{22} = T_4 = \alpha^{13}$, $A_{11} = T_2 = 0$. The first error locator that it will discover is $X_1 = \alpha^{14}$. Then, from (8),

$$e_1 = \frac{|M_2|}{X_1^3(X_1^2 + \sigma_{d1}X_1 + \sigma_{d2})(A_{11}X_1^2 + A_{22})} = \frac{\alpha^6}{\alpha^{12}(\alpha^{13} + \alpha\cdot\alpha^{14} + \alpha^{10})\alpha^{13}} = \alpha^4.$$

Similarly, when it discovers $X_2 = \alpha^{11}$,

$$e_2 = \frac{\alpha^6}{\alpha^3(\alpha^7 + \alpha\cdot\alpha^{11} + \alpha^{10})\alpha^{13}} = \alpha.$$

Then it can solve for d_1, d_2 as before

B. Remarks III

The procedure just described for determining error values is clearly applicable in principle to the determination of erasure values. In this case, however, $\mathbf{s}_d$ must be replaced by ${}_{k_0}\mathbf{s}_d$, the vector of elementary symmetric functions of the $s-1$ erasures other than the one being considered, and the original modified cyclic parity checks T_n by the modified cyclic parity checks defined on the other $s-1$ erasure locators. This means that the determinants appearing in (9), as well as $|M_t|$, must be recomputed to solve for each erasure, in contrast to the solution for the error values; this promises to be tedious and to militate against this method in practice. We mention this possibility only because it does allow calculation of the correct value of an erasure, given only the number of errors and the positions of the other erasures, without knowledge of the location or value of the errors, a capability that might be useful in some application.

The erasure-correction scheme with no errors described previously can be seen to be a special case of this algorithm.

Continued development of the point of view in this paper gives us an alternative method of locating the errors. If we tentatively considered a received symbol as an erasure, in a received word with t errors, then the resulting word would have t errors if the trial symbol were correct, and $t-1$ errors if the trial symbol were in error. The vanishing of the $t \times t$ determinant $|M_t'|$ formed from the T_n'' defined now by $s+1$ erasure locators would then indicate the error locations. The reader may verify the fact that if X_{j_0} is the locator of the trial symbol, $T_n'' = T_{n+1} - X_{j_0}T_n$, and

$$M_t'' = \begin{bmatrix} T_{2t_0-1} - X_{j_0}T_{2t_0-2} & T_{2t_0-2} - X_{j_0}T_{2t_0-3} & \cdots & T_{2t_0-t} - X_{j_0}T_{2t_0-t-1} \\ T_{2t_0-2} - X_{j_0}T_{2t_0-3} & T_{2t_0-3} - X_{j_0}T_{2t_0-4} & \cdots & T_{2t_0-t-1} - X_{j_0}T_{2t_0-t-2} \\ \vdots & \vdots & & \vdots \\ T_{2t_0-t} - X_{j_0}T_{2t_0-t-1} & T_{2t_0-t-1} - X_{j_0}T_{2t_0-t-2} & \cdots & T_{2t_0-2t+1} - X_{j_0}T_{2t_0-2t} \end{bmatrix}.$$

If we expand $|M_t''|$ by columns, many of the resulting determinants will have one column equal to $-X_{j_0}$ times another. The only ones that will not are

$$D_0 = |\mathbf{T}_{(2t_0-1,2t_0-t)}, \mathbf{T}_{(2t_0-2,2t_0-t-1)}, \cdots, \mathbf{T}_{(2t_0-t,2t_0-2t+1)}|$$

$$-X_{j_0}D_1 = |\mathbf{T}_{(2t_0-1,2t_0-t)}, \cdots, \mathbf{T}_{(2t_0-t+1,2t_0-2t+2)}, -X_{j_0}\mathbf{T}_{(2t_0-t-1,2t_0-2t)}|$$

$$X_{j_0}^2D_2 = |\mathbf{T}_{(2t_0-1,2t_0-t)}, \cdots, \mathbf{T}_{(2t_0-t+2,2t_0-2t+3)}, -X_{j_0}\mathbf{T}_{(2t_0-t,2t_0-2t+1)}, -X_{j_0}\mathbf{T}_{(2t_0-t-1,2t_0-2t)}|,$$

and so forth. Thus if X_{j_0} is a root of the polynomial

$$D(X_{j_0}) = \sum_{i=0}^{t} D_i(-X_{j_0})^i,$$

$|M_t''|$ is zero and X_{j_0} is an error locator. It can be verified that $D_i = \sigma_{e(t-i)}D_t$, so that $D(X) = D_t\sigma_e(X)$, and this method is entirely equivalent to the former one. Furthermore, it is clear that

$$D(X) = \begin{vmatrix} X^t & T_{2t_0-1} & T_{2t_0-2} & \cdots & T_{2t_0-t} \\ X^{t-1} & T_{2t_0-2} & T_{2t_0-3} & \cdots & T_{2t_0-t-1} \\ \vdots & \vdots & \vdots & & \vdots \\ X & T_{2t_0-t} & T_{2t_0-t-1} & \cdots & T_{2t_0-2t+1} \\ 1 & T_{2t_0-t-1} & T_{2t_0-t-2} & \cdots & T_{2t_0-2t} \end{vmatrix}.$$

The condition of the vanishing of this matrix determinant is the generalization to the nonbinary case of the 'direct method' of Chien [7]. It appears to offer no advantages in practice, for to get the coefficients of $D(X)$ one must find the determinants of $t+1$ $t \times t$ matrices, whereas the coefficients of the equivalent $\sigma_e(X)$ can be obtained as a by-product of the determination of t.

VI. Implementation

We shall now consider in more detail how a BCH decoder might be realized as a special-purpose computer. Bartee and Schneider [12] and Peterson [8] have discussed the construction of finite-field arithmetic units. We shall assume the availability of an arithmetic unit able to realize, in approximate order of complexity, the following functions of finite-field elements: addition ($X = X_1 + X_2$), squaring ($X = X_1^2$), multiplication by β_m, $m_0 \leq m \leq m_0 + d - 2$ ($X = \beta^m X_1$), inversion ($X = X_1^{-1}$), and multiplication ($X = X_1X_2$). Furthermore, because of the bistable nature of common computer elements, we shall assume $p = 2$, so that subtraction is identical to addition

and squaring is linear. Finally, if the fields on which the symbol locators and symbol values are defined are different ($M \neq N$), we shall assume that all elements of the smaller field are converted to their representations in the larger field of which the smaller field is a subfield, and that all operations are carried out in the larger field, say, $GF(2^N)$.

We attempt to estimate the approximate complexity of the algorithms described earlier by estimating the number of multiplications required by each and the number of memory registers. All registers will, of course, be N bits long. In a straightforward serial multiplication, the amount of time needed for one multiplication is also proportional to N [8].

During the computation, the received sequence of symbols must be stored in some buffer, awaiting correction. Once the S_m and Y_k have been determined, no further access to this sequence is required.

The calculation of the parity checks $S_m \equiv r(\beta^m) = r_1\beta^{m(n_0-1)} + r_2\beta^{m(n_0-2)} + \cdots + r_{n_0}$ is accomplished by the iteration

$$S_m = ((r_1\beta^m + r_2)\beta^m + r_3)\beta^m + r_4 \cdots$$

which involves $n_0 - 1$ multiplications by β^m. $d - 1$ such parity checks must be formed, requiring $d - 1$ memory registers.

δ_d can be calculated at the same time. We note that $\sigma_{dk} = {}_{k_0}\sigma_{dk} + Y_{k_0}{}_{k_0}\sigma_{d(k-1)}$; δ_d can be calculated by this recursion relation as each new Y_k is determined. Adding a new Y_k requires s' multiplications when s' are already determined, so that the total number of multiplications, given s erasures, is $s - 1 + s - 2 + \cdots = \binom{s}{2} < \frac{1}{2}d^2$. s memory registers are required ($\sigma_{d0} = 1$).

The modified cyclic parity checks T_n are then calculated by (3). Each requires s multiplications, and there are $d - s - 1$ of them, so that this step requires $s(d - s - 1) < \frac{1}{4}d^2$ multiplications and $d - s - 1$ memory registers.

Equation (6) is then set up in $t_0(t_0 + 1) < \frac{1}{4}d^2$ memory registers. In the worst case, $t = t_0$, the reduction to upper triangular form of these equations will require t_0 inversions and

$$t_0(t_0 + 1) + (t_0 - 1)t_0 + \cdots + (1)(2) = \begin{bmatrix} 2t_0 + 2 \\ 3 \end{bmatrix} + \begin{bmatrix} t_0 + 1 \\ 2 \end{bmatrix} < \frac{(t_0 + 1)^3}{3}$$

multiplications. As d becomes large, this step turns out to be the most lengthy, requiring as it does $\sim d^3/24$ multiplications.

Determination of δ_e from these reduced equations involves, in the worst case, a further $\binom{t_0}{2} < d^2/8$ multiplications, and t_0 memory registers.

Finding the roots of $\sigma_e(X) = \sum_{i=0}^{t} \sigma_{e(t-i)}X^i$ is best accomplished by the method of Chien [7]. If $\sum_{i=0}^{t} \sigma_{e(t-i)} = 0$, then 1 is a root of $\sigma_e(X)$. Use the special multipliers by β^m in the arithmetic unit, and let $\sigma'_{e(t-i)} = \beta^{m_0+t-i}\sigma_{e(t-i)}$. Now $\sum_{i=0}^{t} \sigma'_{e(t-i)} = \beta^{m_0+t} \sum_{i=0}^{t} \beta^{-i}\sigma_{e(t-i)}$, which will be zero when $\beta^{-1} = \beta^{n_0-1}$ is a root of $\sigma_e(X)$. All error locators can therefore be found with n_0t multiplications by β^m, and stored in t memory registers.

At this point we have the option of solving for the error values directly, by (8), or indirectly, by treating the errors as erasures and using (7).

If we choose the former method, we need the t $(t - 1) \times (t - 1)$ determinants A_{jj} of (9). In general this requires $\frac{1}{4}t\binom{2t}{3} < t^4/3$ multiplications, which is rapidly too many as t becomes large. There is a method of calculating all A_{jj} at once which seems feasible for moderate values of t. Let $B_{a_1, a_2,\ldots,a_j}$ be the determinant of the $j \times j$ matrix which remains when all of the rows and columns but the a_1th, a_2th, $\cdots$, a_jth are struck from M_t. In this notation $|M_t| = B_{1,2,\ldots,t}$ and $A_{jj} = B_{1,2,\cdots,j-1,j+1,\cdots,t}$. The reader may verify the fact that

$$B_{a_1,a_2,\cdots,a_j} = T_{2t_0-2a_j}B_{a_1,a_2,\cdots,a_{j-1}} + T^2_{2t_0-2a_j+1}B_{a_1,a_2,\cdots,a_{j-2}} + T^2_{2t_0-2a_j+2}B_{a_1,a_2,\cdots,a_{j-2},a_{j-1}} + \cdots$$

by expanding B in terms of the minors of its last row and cancelling those terms which, because of symmetry, appear twice. The use of this recursion relation allows calculation of all A_{jj} with N_t multiplications (not counting squares), where, for small t, N_t is $N_2 = 0$ (see Example 3), $N_3 = 3$, $N_4 = 15$, $N_5 = 38$, $N_6 = 86$, $N_7 = 172$, $N_8 = 333$, and $N_9 = 616$.

Once the A_{jj} are obtained, the denominator of (8) can be expressed as a single polynomial $E(X)$ by st multiplications; $E(X)$ has terms in X^m for all m such that $m_0 + 2t_0 - 2t \leq m \leq m_0 + 2t_0 + s$, or a total of $2t + s + 1$ terms. The value of $E(X)$ can therefore be obtained for $X = 1, \beta^{-1}, \beta^{-2}, \cdots$, in turn by the Chien method of solving for the roots of $\sigma_e(X)$, and in fact these two calculations may be done simultaneously. Whenever β^{n_0-i} is a root of $\sigma_e(X)$, $E(\beta^{n_0-i})$ will appear as the current value of $E(X)$. Since $|M_t|$ will have been obtained as a by-product of solving for $\sigma_e(X)$, an inversion and multiplication will give the error value corresponding to $X_{i_0} = \beta^{n_0-i}$. An additional $n_0(s + 2t)$ multiplications by β^m are involved here, and $s + 2t$ memory registers.

Finally, we have only the problem of solving for s erasure values, if we have already determined the error values as above, or of $s + t$ erasures, if we know only the error locators. We use (7), which requires the elementary symmetric functions of all erasure locators but one. The reader may verify the fact that ${}_{k_0}\sigma_{dk} = Y^{-1}_{k_0}(\sigma_{d(k+1)} - {}_{k_0}\sigma_{d(k+1)})$. Beginning with ${}_{k_0}\sigma_{d(s-1)} = Y^{-1}_{k_0}\sigma_{ds}$, we can find all ${}_{k_0}\sigma_{dk}$ from the σ_{dk} with $s - 1$ multiplications and an inversion. Then the calculation of (7) requires $2(s - 1)$ multiplications and an inversion. Doing this s times, to find all erasure values requires, therefore, $3s(s - 1)$ multiplications and s inversions. Or we can alter $s - 1$

parity checks after finding the value of the first erasure, and repeat with $s' = s - 1$, and so forth; under the assumption that all $Y_{k_0}^m$ are readily available, this alternative requires only $2s(s-1)$ multiplications and s inversions. Thus our procedure is simpler than the general methods for solving s linear equations in s unknowns, which require a number of multiplications proportional to s^3.

In order to compare the alternative methods of finding error values, we simply compare the number of multiplications needed in each case, leaving aside all analysis of any other equipment or operations needed to realize either algorithm. For the first method, we need approximately N_t multiplications to find the error values, and $2s(s-1)$ to find the erasures; for the second, $2(s+t)(s+t-1)$ to find both the erasures and the errors. Using the values of N_t given here, we find that the first method definitely requires fewer multiplications when $t \leq 7$, which suggests that it ought to be considered whenever the minimum distance of the code is 15 or less.

To summarize, we note that there are two steps with a number of multiplications by β^m proportional to n_0 and a step with a number of memory registers proportional to d^2, and a buffer with memory proportional to n_0; and the complexity of a multiplication and the length of a register are both proportional to N. It follows that if $d \sim \delta n_0$ and $N \sim \log_2 n_0$, we can estimate that the complexity of a BCH decoder will increase with code length n_0 as n_0^b, where b is some small power of the order of 3. Therefore we can expect that implementation of a decoder for a long and complicated BCH code will not be prohibitively more difficult than for a short, simple code.

Acknowledgment

The author is indebted to W. W. Peterson, from whose book [8] comes most of his knowledge of this field. These results appeared in the author's doctoral thesis [13], during the research for which he was supported by a National Science Foundation Graduate Fellowship.

References

[1] R. C. Bose, and D. K. Ray-Chaudhuri, "On a class of error-correcting binary group codes," *Inform. and Contr.*, vol. 3, pp. 68–79, March 1960.
[2] ——, "Further results on error-correcting binary group codes," *Inform. and Contr.*, vol. 3, pp. 279–290, September 1960.
[3] A. Hocquenghem, "Codes correcteurs d'erreurs," *Chiffres*, vol. 2, pp. 147–156, September 1959.
[4] W. W. Peterson, "Encoding and error-correction procedures for the Bose-Chaudhuri codes," *IRE Trans. on Information Theory*, vol. IT-6, pp. 459–470, September 1960.
[5] T. C. Bartee and D. I. Schneider, "An electronic decoder for Bose-Chaudhuri-Hocquenghem error-correcting codes," *IRE Trans. on Information Theory*, vol. IT-8, pp. S17–S24, September 1962.
[6] D. Gorenstein and N. Zierler, "A class of cyclic linear error-correcting codes in p^m symbols," *J. SIAM*, vol. 9, pp. 207–214, June 1961.
[7] R. T. Chien, "Cyclic decoding procedures for Bose-Chaudhuri-Hocquenghem codes," *IEEE Trans. on Information Theory*, vol. IT-10, pp. 357–363, October 1964.
[8] W. W. Peterson, *Error-Correcting Codes*, New York: M.I.T. Press-Wiley, 1961.
[9] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *J. SIAM*, vol. 8, pp. 300–304, June 1960.
[10] P. Elias, "Coding for two noisy channels," in *Information Theory*, C. Cherry, Ed., New York: Academic Press, 1956.
[11] R. T. Chien and D. T. Tang, "On detecting errors after correction," *Proc. IEEE*, vol. 62, p. 974, August 1964.
[12] T. C. Bartee and D. I. Schneider, "Computation with finite fields," *Inform. and Contr.*, vol. 6, pp. 79–98, June 1963.
[13] G. D. Forney, Jr., "Concatenated codes," Sc.D. Thesis, Dept. of Electrical Engineering, M.I.T., Cambridge, Mass., June, 1965. To appear as Technical Report 440, M.I.T. Research Laboratory of Electronics, Cambridge, Mass.