X-711-67-431

NASA TM X- 55947

INTRODUCTION TO BOSE-CHAUDHURI-HOCQUENGHEM CODES

S. A. TRETTER

GPO PRICE $ ____________

CFSTI PRICE(S) $ ____________

Hard copy (HC) 3.00

Microfiche (MF) .65

ff 653 July 65

SEPTEMBER 1967

GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND

N67-39350	
(ACCESSION NUMBER)	(THRU)
39	1
(PAGES)	(CODE)
TMX-55947	07
(NASA CR OR TMX OR AD NUMBER)	(CATEGORY)

FACILITY FORM 602

X-711-67-431

INTRODUCTION TO BOSE-CHAUDHURI-HOCQUENGHEM CODES

by

S. A. Tretter*

September 1967

*Dr. S. A. Tretter is an Assistant Professor in Electrical Engineering at the University of Maryland, College Park, Md. This work was performed while he was a member of The Goddard 1967 Summer Workshop.

Goddard Space Flight Center
Greenbelt, Maryland

CONTENTS

AN INTRODUCTION TO BOSE-CHAUDHURI-HOCQUENGHEM CODES

I. INTRODUCTION

Bose-Chaudhuri-Hocquenghem codes are the most efficient class of algebraic, block codes known for correcting random errors. These codes encompass a wide range of rate and error-correcting capability. They were discovered by Hocquenghem [14] in 1959 and independently by Bose and Chaudhuri [6, 7] in 1960 as a constructive proof that binary block codes of length $2^m - 1$ exist that correct t errors with at most mt parity check symbols. The Reed-Solomon, Golay, and well known Hamming codes belong to this class. The first decoding procedure for binary codes was discovered by Peterson [17] in 1960. A generalized method for decoding both binary and non-binary codes was found soon afterwards by Gorenstein and Zieler [25, 26]. By taking advantage of the cyclic nature of BCH codes, Chien [9] in 1964 proposed a decoding procedure for binary codes resulting in increased speed and decreased complexity for special purpose decoding computers and in 1965 Massey [15] discovered a step-by-step procedure for decoding both binary and non-binary codes that has slightly simpler hardware mechanization than Chien's method. The decoding methods were extended to include erasures as well as errors by Forney [11] thus improving the performance of the algebraic codes relative to the optimum probabilistic decoding procedures. The principal advantage of the algebraic block codes is the simplicity of the coding and decoding algorithms and the resulting efficiency of implementation.

II. FUNDAMENTAL MATHEMATICAL CONCEPTS

This report is designed to be a tutorial introduction to the BCH codes. These codes are based on the concepts of modern algebra. In this section the basic mathematical concepts and definitions necessary for understanding and using these codes are introduced. The reader desiring a more complete and rigorous presentation should consult Peterson [18] and texts on modern algebra.

Concept 1 arithmetic modulo p

Given two numbers, b and p, if b is divided by p, the result is a quotient q and a remainder r with r less than p. In other words, b can be expressed as:

$$b = qp + r$$

The number $c = b$ modulo p, usually written as $c = b \bmod p$, is defined to be the remainder r when b is divided by p.

Example 1.

$$21 \bmod 5 = 1$$

$$\text{since } 21 = 4 \times 5 + 1$$

Definition 1. Equivalence classes

Any numbers having the same remainders with respect to p, are said to be equivalent or in the same equivalence class. That is, $a = b$ if $a \bmod p = b \bmod p$.

Example 2.

All even numbers modulo 2 are equivalent to 0. All odd numbers modulo 2 are equivalent to 1.

Example 3. Addition Modulo 2

$$(0 + 0) \bmod 2 = 0$$

$$(0 + 1) \bmod 2 = 1$$

$$(1 + 1) \bmod 2 = 0$$

Observe that subtraction modulo 2 is equivalent to addition modulo 2 since $1 - 1 = 0$.

Comment: Examples 2 and 3 illustrate the relationship of the binary number system to arithmetic modulo 2.

Concept 2 Polynomials modulo $g(x)$

Definition 2. Polynomial of Order n

A polynomial or order n is an algebraic expression of the form $f(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$ where $a_0, a_1, \ldots, a_n$ are fixed numbers and x is an indeterminate.

Given two polynomials f (x) and g (x) the result of dividing f (x) by g (x) is a quotient q (x) and a remainder r (x) of order less than g (x). That is, f (x) can be written as:

$$f(x) = q(x)\,g(x) + r(x)$$

Definition 3. f (x) mod g (x)

f (x) mod g (x) is defined to be the remainder r (x). Polynomials having the same remainder are said to be equivalent or in the same equivalence class.

Example 4. $(x^3 + x^2 + 1) \bmod (x + 1)$

$$\begin{array}{rr} & x^2 \\ x+1 & \overline{\big)\, x^3 + x^2 + 1} \\ & \underline{x^3 + x^2 \phantom{{}+1}} \\ & 1 \end{array}$$

$$q(x) = x^2,\ r(x) = 1$$

so that

$$x^3 + x^2 + 1 = x^2 (x + 1) + 1$$

and

$$(x^3 + x^2 + 1) \bmod (x + 1) = 1$$

Concept 3 Group

Definition 4. Group

A group is a collection of elements $a_1, a_2, a_3, \ldots$ and an operation denoted by $\cdot$ such that the following axioms are satisfied:

1. Closure

For any two elements a_1, and a_2, in the group $a_1 \cdot a_2$ is in the group.

2. Associative Law

$$a_1 \cdot (a_2 \cdot a_3) = (a_1 \cdot a_2) \cdot a_3$$

3. Identity element

The group contains a unique element I such that $a . I = I . a = a$ for all a in the group.

4. Inverses

For each element a_1, the group contains an inverse element a_1 such that

$$a_1 \cdot a_1^{-1} = a_1^{-1} \cdot a_1 = I$$

If for any two elements $a_1 . a_2 = a_2 . a_1$ the group is called commutative or Abelian. The groups used in describing algebraic codes are Abelian.

Example 5. Additive group of integers modulo 5

Let the elements of the group be the integers 0, 1, 2, 3, 4, and the operation . be addition modulo 5 denoted by the symbol $\oplus$. Then the identity element I = 0 since $0 \oplus a = a$. The addition table and the table of inverses are shown below.

Addition Table

	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

Table of Inverses

a	-a
0	0
1	4
2	3
3	2
4	1

Example 6. Multiplicative group of integers modulo 5

Let the group elements be 1, 2, 3, 4, and the operation . be multiplication with the result reduced modulo 5. Then the identity I = 1.

Multiplication Table

	1	2	3	4
1	1	2	3	4
2	2	4	1	3
3	3	1	4	2
4	4	3	2	1

Table of Inverses

a	a^{-1}
1	1
2	3
3	2
4	4

Note: It can be shown [18] that the integers 1, 2, . . . , p^{-1}, where p is a prime number, and the operation, multiplication modulo p , form a group.

Definition 5: Order of a Group

The order of a group is the number of elements in the group.

Definition 6: Powers of elements

$$a^0 = I$$

$$a^2 = a \cdot a$$

$$a^3 = a \cdot a \cdot a = a^2 \cdot a$$

etc.

Definition 7. Order of an element

The order of an element is the smallest nonzero integer e such that $a^e = I$. It can be shown that the order of an element always divides the order of the group.

Example 7.

The order of the group in Example 6 is 4. The order of the various elements will now be determined.

$$1^1 = 1 \text{ so } e_1 = 1$$

$$2^2 = 4,\ 2^3 = 8 = 3,\ 2^4 = 16 = (3 \times 2) = 1 \text{ so } e_2 = 4$$

$$3^2 = 9 = 4,\ 3^3 = 12 = 2,\ 3^4 = 6 = 1 \text{ so } e_3 = 4$$

$$4^2 = 16 = 1 \text{ so } e_4 = 2$$

Thus the order of each element divides 4, the order of the group. Notice that the powers of the elements 2 and 3 generate all the other elements of the group.

Definition 8: Primitive element

An element, a , whose powers generate all the group elements is called a primitive element. It can be shown that every group contains at least one primitive element.

Definition 9: Subgroup

A subgroup is a set of elements taken from a group satisfying all the group axioms.

Example 8

The elements 1, 4 form a subgroup of the group in Example 6.

Concept 4 Field

Definition 10: A field is a set of elements closed under addition (+) and multiplication (.) which satisfy the following axioms:

1. The set of elements is an Abelian group under addition.
2. The set of nonzero elements form an Abelian multiplicative group.
3. The distributive law applies: $a(b+c) = ab + ac$

Example 8

1. The real numbers form a field with ordinary arithmetic.
2. The numbers 0, 1 form a field under arithmetic modulo 2.
3. The numbers 0, 1, 2, 3, 4, form a field under arithmetic modulo 5.
4. It can be shown that the integers 0, 1, . . . , p^{-1} where p is a prime number form a field under arithmetic modulo p.

Definition 11: Galois Field

A field containing a finite number of elements p is called a Galois field. These will be denoted as GF (p).

Concept 5 Polynomial over a field

Definition 12: An expression of the form

$$f_n x^n + f_{n-1} x^{n-1} + \cdots + f_0$$

is called a polynomial of order n over GF (p) if the coefficients $f_0, f_1, \ldots, f_n$ are all elements of GF (p) and $f_n \neq 0$. GF (p) is called the ground field.

Addition and multiplication of polynomials are performed according to the ordinary rules except that the coefficients are found using arithmetic modulo p. Therefore if

$$f(x) = \sum_{i=0}^{n} f_i x^i$$

and

$$g(x) = \sum_{i=0}^{n} g_i x^i$$

$$f(x) + g(x) = \sum_{i=0}^{n} [(f_i + g_i) \bmod p] x^i$$

$$f(x) g(x) = \sum_{i=0}^{2n} \left[\left(\sum_{j=0}^{i} f_j g_{i-j} \right) \bmod p \right] x^i$$

Example 9. Polynomials over GF (2)

Let the ground field be GF (2) with elements 0 and 1. Let $f(x) = x^2 + 1$ and $g(x) = x^3 + x^2 + 1$. Then $f(x) + g(x) = x^3 + (1+1) x^2 + (1+1) = x^3$ and

$$\begin{array}{rr}
 & x^3 + x^2 + 1 \\
 & x^2 + 1 \\
\hline
 & x^5 + x^4 + x^2 \\
 & x^3 + x^2 + 1 \\
\hline
f(x)\, g(x) = & x^5 + x^4 + x^3 + 1
\end{array} .$$

Definition 13. Irreducible Polynomial

A polynomial f (x) is irreducible over GF (p) if it can not be expressed as the product of two polynomials g (x) and h (x) each of degree at least one, with coefficients in GF (p).

Example 10:

In GF(2) $x^2 + 1 = (x+1)^2$ is not irreducible while $x^2 + x + 1$ cannot be expressed as the product of two polynomials over GF (2) and is therefore irreducible.

Comment: Given a polynomial $g(x)$ over $GF(p)$ of order n and a polynomial $f(x)$ over $GF(p)$ of arbitrary order, $f(x) \bmod g(x)$ was defined in Concept 2 to be the remainder when $f(x)$ is divided by $g(x)$. Therefore $f(x) \bmod g(x)$ is the $n - 1$ order polynomial $r(x) = r_{n-1} x^{n-1} + \ldots + r_0$.

For fixed $g(x)$ and arbitrary $f(x)$, $r(x)$ can be p^n different polynomials since $r_0, r_1, \ldots, r_{n-1}$ can each be any one of the p elements of $GF(p)$. It can be seen that these p^n polynomials form an additive group. If $g(x)$ is irreducible it can be shown [18] that the $p^n - 1$ nonzero polynomials form a multiplicative group if multiplication is performed modulo $g(x)$. Therefore under the operations of polynomial addition and polynomial multiplication modulo $g(x)$ with all coefficients determined using arithmetic modulo p, the p^n polynomials of order $n - 1$ form a field when $g(x)$ is irreducible. Therefore these p^n polynomials are the elements of a Galois field $GF(p^n)$. It is convenient for polynomial addition to represent the elements $r(x) = r_{n-1} x^{n-1} + \ldots + r_0$ of $GF(p^n)$ by the n dimensional vector $r = [r_0, r_1, \ldots r_{n-1}]$. This notation will be used interchangeably with the polynomial form.

Example 11.

Let the base field be $GF(2)$ and $g(x) = x^3 + x + 1$. It can be shown that $g(x)$ is an irreducible polynomial so that the field generated by polynomials modulo $g(x)$ has order $2^3 = 8$. It is instructive to examine the elements $x^k \bmod g(x)$ for $k = 0, 1, \ldots$. The table below was generated by dividing x^k by $g(x)$ to find the remainder $r(x)$.

	r_0	r_1	r_2
0	0	0	0
1	1	0	0
x	0	1	0
x^2	0	0	1
x^3	1	1	0
x^4	0	1	1
x^5	1	1	1
x^6	1	0	1
x^7	1	0	0

As stated previously the nonzero elements of $GF(2^3)$ must form a multiplicative group of order 7. From the table it can be seen that the powers of the element x generate all the elements of this group. Therefore x is a primitive element. The polynomial $g(x)$ is called a primitive polynomial. Also observe that $x^7 = 1$.

It can be shown that for any element, a, of a group of order e, $a^e = 1$. To find the inverse element of any member of the group observe that $x^i x^j = x^{i+j} = 1$ if $i + j = 7$ so that $[x^i]^{-1} = x^{-i} = x^j$. For example, from the table it is observed that $x^3 = 1 + x$ and $x^4 = x + x^2$ so that $x^3 . x^4$ should be 1. Multiplying gives:

$$\begin{array}{rlll} & 1+x & & \\ & \underline{x+x^2} & & \\ & x+x^2 & & \\ & & \underline{x^2+x^3} & \\ x^3\cdot x^4 = x & & +x^3 & \end{array}$$

and reducing $x + x^3$ modulo $g(x)$

$$\begin{array}{rl} & \quad 1 \\ x^3+x+1 \,\big) & \overline{x^3+x} \\ & \underline{x^3+x+1} \\ & 1 \end{array}$$

shows that the remainder is 1 as expected.

Concept 6 Roots and Extension Fields

Definition 14: Root of a Polynomial

Given a polynomial $f(x)$, any element α such that $f(\alpha) = 0$ is called a root of $f(x)$.

If α is a root of $f(x)$ then $x - \alpha$ must be a factor of $f(x)$, i.e., $f(x) = (x - \alpha) h(x)$.

The roots of a polynomial over GF(p) may or may not belong to GF(p). This fact is illustrated by the following two examples.

Example 12. Polynomial over GF (2) with roots in GF (2)

Let $f(x) = x^2 + 1 = (x + 1)^2$. Then clearly $f(1) = 0$ and 1 is a double root of $f(x)$ and belongs to GE(2).

Example 13. Polynomial over real numbers with no real roots

Let $f(x) = x^2 + 1$, then there is no real number, α, such that $\alpha^2 + 1 = 0$. However, if our root field is extended to include complex numbers, then a root α of $f(x)$ is customarily denoted as $\alpha = i = \sqrt{-1}$. Since α is a root of $f(x)$,

$$f(\alpha) = \alpha^2 + 1 = 0$$

$$\text{or } \alpha^2 = -1$$

$$\text{and } \alpha^3 = -\alpha$$

$$\text{and } \alpha^4 = -\alpha^2 = 1$$

It should be observed that any complex number $c = a + i b$ is a linear combination of the powers of the roots of $f(x)$. That is, $c = a\alpha^0 + \alpha b$.

Example 14. Polynomial over GF (2) without roots in GF(2)

Let $f(x) = x^3 + x + 1$ as in Example 11. Since $f(1) = f(0) = 1$, $f(x)$ has no roots in GF(2). Let a root of $f(x)$ in some extension field be abstractly designated as α. Therefore

$$f(\alpha) = \alpha^3 + \alpha + 1 = 0$$

$$\text{or } \alpha^2 = \alpha + 1$$

Using this relationship for α^3, α^k for $k = 0, 1, \cdots$ can always be expressed as a linear combination of $\alpha^0 = 1$, α and α^2. For example

$$\alpha^4 = \alpha \cdot \alpha^3 = \alpha^2 + \alpha$$

$$\alpha^5 = \alpha \cdot \alpha^4 = \alpha^3 + \alpha^2 = (\alpha + 1) + \alpha^2 = \alpha^2 + \alpha + 1 \text{ etc.}$$

It is convenient to represent $\alpha^k = c_0 + c_1\alpha + c_2\alpha^2$ as a 3 dimensional vector $\alpha^k = [c_0, c_1, c_2,]$ in some cases. A table of the powers of the root α of $f(x)$ is shown on the following page.

Table

	c_0	c_1	c_2
α^0	1	0	0
α^1	0	1	0
α^2	0	0	1
α^3	1	1	0
α^4	0	1	1
α^5	1	1	1
α^6	1	0	1
α^7	1	0	0

The elements $1, \alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6$, form a group under multiplication. Since $\alpha^7 = 1$, $\alpha^i \alpha^j = \alpha^{i+j} = 1$ if $i + j = 7$. Therefore $[\alpha^i]^{-1} = \alpha^j$ so that each element has an inverse. For any integer k, $k = q \times 7 + k \bmod 7$ so $\alpha^k = \alpha^{q \times 7} \alpha^{k \bmod 7} = \alpha^{k \bmod 7}$. Therefore the set of powers of α is closed. From the table it can be seen that the powers of the root α generate all the elements of the group so that α is a primitive element.

Definition 15. Primitive Polynomial

A primitive polynomial is an irreducible polynomial with at least one primitive root α.

It can easily be shown that all linear combinations of 1, α, and α^2, i.e., elements of the form $c_0 + c_1 \alpha + c_2 \alpha^2$ with the coefficients c_0, c_1, c_2 taken from GF (2), form an additive group with $2^3 = 8$ elements. Therefore the roots generate an extension field GF(2^3).

The reader should compare Example 14 with Example 11. These examples illustrate the fact that the field corresponding to polynomials modulo $x^3 + x + 1$ is identical with the extension field generated by the root α of $x^3 + x + 1$. This equivalence is true in general. Given an irreducible polynomial $g(x)$ of degree n over GF(p) with a root α an arbitrary polynomial $f(x)$ can be expressed on applying the Euclidean division algorithm as

$$f(x) = q(x)\, g(x) + r(x)$$

where the degree of $r(x)$ is less than n.

Substituting α for x gives:

$$f(\alpha) = q(\alpha)\, g(\alpha) + r(\alpha)$$

but since α is a root of $g(x)$, $g(\alpha) = 0$ and $f(\alpha) = r(\alpha)$ which is exactly equivalent to $f(x) \bmod g(x) = r(x)$ with x replaced by α. Therefore either point of view generates the Galois field GF (p^n).

The result stated in the following theorem is important because it determines the block length of the algebraic codes.

Theorem 1.

A primitive polynomial $g(x)$ of order n over GF (p) divides $x^{p^n-1} - 1$.

Proof:

Any root α_i of $g(x)$ generates an extension field of GF (p) with p^n elements. The set of $p^n - 1$ nonzero elements form a multiplicative group. Therefore $\alpha_i^{p^n-1} = 1$ or $\alpha_i^{p^n-1} - 1 = 0$ for $i = 1, \ldots, n$ since $p^n - 1$ is the order of the multiplicative group. Thus each root of

$$g(x) = \prod_{i=1}^{n} (x - \alpha_i)$$

is also a root of $x^{p^n-1} - 1$ so $g(x)$ must divide $x^{p^n-1} - 1$.

QED

If $g(x)$ is irreducible but not primitive, its roots will generate an extension field of order $e < p^n$ so that $g(x)$ will divide $x^{e-1} - 1$. These polynomials are useful for generating codes of length $e - 1$. It should be observed that $g(x)$ will still divide $x^{p^n-1} - 1$ since its roots form a subfield of GF(p^n).

The following theorem and corollary are important in finding the generator polynomials (which will be defined later) for BCH codes.

Theorem II

Given a polynomial

$$f(x) = \sum_{i=0}^{n} f_i x^i$$

over GF (p),

$$[f(x)]^p = f(x^p).$$

Proof:

First consider the case where $n = 1$. Then according to the binomial theorem

$$(f_0 + f_1 x)^p = \sum_{r=0}^{p} \binom{p}{r} f_0^r (f_1 x)^{p-r} = f_0^p + f_1^p x^p$$

since

$$\binom{p}{r} = \frac{p!}{r!\,(p-r)!}$$

is divisible by p except when $r = 0$ or p. Since the coefficients f_i are elements of GF (p), $f_i^p = f_i$ so that

$$(f_0 + f_1 x)^p = f_0 + f_1 x^p .$$

Now let $f(x)$ be an n^{th} order polynomial and assume the theorem is true for $n - 1^{th}$ order polynomials. Then

$$f(x) = \sum_{\lambda=0}^{n} f_i x^i = \sum_{i=0}^{n-1} f_i x^i + f_n x^n$$

and

$$[f(x)]^p = \left[\sum_{i=0}^{n-1} f_i x^i + f_n x^n \right]^p$$

$$= \left[\sum_{i=0}^{n-1} f_i x^i \right]^p + f_n^p x^{pn}$$

$$= \sum_{i=0}^{n} f_i (x^p)^i = f(x^p)$$

Q.E.D.

Corollary 1.

If α is a root of the polynomial $f(x)$ over GF(p), then α^p is also a root.

Proof:

According to Theorem II

$$[f(x)]^p = f(x^p)$$

so that

$$[f(\alpha)]^p = 0 = f(\alpha^p)$$

Q.E.D.

Definition 16: Monic Polynomial

A polynomial with unity as the coefficient of its highest order term is called a monic polynomial.

Definition 17: Minimum Polynomial

Let β be an element of an extension field GF(p^n). The monic polynomial $m(x)$ of smallest degree over GF(p) such that $m(\beta) = 0$ is called the minimum polynomial of β. It can be shown [18] that the degree of $m(x)$ is n or less.

Example 15

Consider the extension field GF(2^3) discussed in Example 14. Since α is a root of $f(x) = x^3 + x + 1$, according to Corollary 1, α^2 and $(\alpha^2)^2 = \alpha^4$ must also be roots of $f(x)$. Notice that $(\alpha^4)^2 = \alpha^8 = \alpha$. Therefore α, α^2, and α^4 are the three roots of $f(x)$ so that $f(x) = (x - \alpha)(x - \alpha^2)(x - \alpha^4)$. Using the table of Example 14,

$$f(\alpha^2) = \alpha^6 + \alpha^2 + 1 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

and

$$f(\alpha^4) = \alpha^{12} + \alpha^4 + 1 = \alpha^5 + \alpha^4 + 1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

The minimum polynomial $m_3(x)$ for α^3 will now be found. According to Corollary 1 α^6, $\alpha^{12} = \alpha^5$, $\alpha^{10} = \alpha^3$ must be roots of $m_3(x)$. Therefore $m_3(x) = (x - \alpha^3)(x - \alpha^6)(x - \alpha^5) = [x^2 - (\alpha^3 + \alpha^6)x + \alpha^9](x - \alpha^5) = x^3 - (\alpha^3 + \alpha^5 + \alpha^6)x^2 + (\alpha^8 + \alpha^9 + \alpha^{11})x - \alpha^{14}$.

According to the table

$$\alpha^3 + \alpha^5 + \alpha^6 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\alpha^8 + \alpha^9 + \alpha^{11} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

and

$$\alpha^{14} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

so that

$$m_3 (x) = x^3 + x^2 + 1$$

It is instructive to find m_3 (x) by an alternate method. Assume that $m_3 (x) = c_0 + c_1 x + c_2 x^2 + x^3$.

Then $m_3 (\alpha^3) = c_0 + c_1 \alpha^3 + c_2 \alpha^6 + \alpha^9 = 0 =$

$$c_0 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + c_1 \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} + c_2 \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

From the last of the three equations above it is concluded that $c_2 = 1$. From the second equation c_1 is found to be 0 and thus c_0 must be 1. Thus $m_3 (x) = x^3 + x^2 + 1$ as before.

III. CIRCUITS FOR ARITHMETIC IN GF(p^m)

Algebraic block codes are attractive because of the simplicity of the digital equipment necessary for coding and decoding. The simplicity in the case of binary codes results from the fact that multiplication of elements in GF(2^m) can be performed instantaneously using logic gates while division is easily performed with shift registers. Some typical circuits used for GF (p^m) arithmetic are discussed in this section.

A. Multiplication of Elements of GF(p^m)

It was demonstrated in Section II that any element of GF(p^m) can be represented as a linear combination of the elements 1, α, α^2, . . . , α^{m-1} where α is the root of a primitive polynomial of order m. In decoding BCH codes it is necessary to find powers and products of elements in GF(p^m). For binary codes this can be performed readily using logic gates. The technique is illustrated for GF (2^3) in the following example.

Example 16.

Let $GF(2^3)$ be represented as in Example 14. Let two elements of $GF(2^3)$ be

$$c = c_0 + c_1 \alpha + c_2 \alpha^2$$

$$d = d_0 + d_1 \alpha + d_2 \alpha^2$$

Then cd can be found using ordinary multiplication to be

$$cd = d_0 c_0 + (d_0 + d_1 c_0) \alpha + (d_0 c_2 + d_1 c_1 + d_2 c_0) \alpha^2$$
$$+ (d_1 c_2 + d_2 c_1) \alpha^3 + d_2 c_2 \alpha^4$$

But from the table in Example 14

$$\alpha^3 = 1 + \alpha \text{ and } \alpha^4 = \alpha + \alpha^2$$

Therefore, after simplification,

$$cd = (d_0 c_0 + d_1 c_2 + d_2 c_1) + (d_0 c_1 + d_1 c_0 + d_1 c_2 + d_2 c_1 + d_2 c_2) \alpha$$
$$+ d_2 c_2 \alpha^2$$

The GF (2) equations for the coefficients of cd can easily be translated into a set of Boolean equations. These equations can be minimized and the corresponding logic network synthesized.

B. Division of f (x) by g (x)

The notation of Peterson (18) will be used here. The symbol ⊕ represents a modulo p adder and the symbol □ represents a storage element of a shift register. The circuit shown in Figure 1 can be used to divide the polynomial $f(x) = f_0 + f_1 x + \ldots + f_n x^n$ by the polynomial $g(x) = g_0 + g_1 x + \ldots + x^m$. The coefficients of f (x) are applied to the input serially from highest order to lowest order. The coefficient of the quotent appear serially at the output from highest to lowest order also. After n shifts the coefficients of the remainder are left in the shift register with the coefficient of highest order on the right. The operation of this circuit is analogous to ordinary long division and is explained in detail in [18] pp. 111-113.

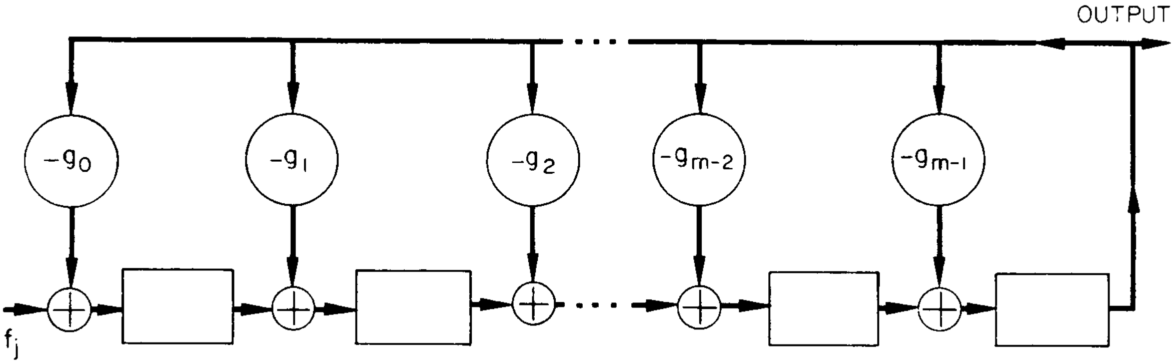

Figure 1–Circuit for Division of Polynomials

For decoding it is necessary to evaluate f (α) where α is a root of $g(x)$. Since

$$f(x) = q(x)\, g(x) + r(x), \quad f(\alpha) = r(\alpha)$$

and therefore only the remainder $r(x)$ is needed. It is instructive to examine the operation of the division circuit from the point of finding f (α). Consider the storage elements of the shift register to contain from left to right the coefficients of 1, $\alpha, \ldots, \alpha^{m-1}$. The storage elements are first set to zero. The contents of the register after the first shift are f_n, after 2 shifts are $f_{n-1} + \alpha f_n$, and after m shifts are $f_{n-m+1} + f_{n-m+2}\ \alpha \ldots + f_n\ \alpha^{m-1}$ since feedback occurs only when the data reaches the last stage of the shift register. At the m + 1 shift feedback begins to occur. Suppose, for the time being, that the feedback link was broken and that the register contained additional storage elements. Then after the m + 1 shift the contents of the register would correspond to

$$f_{n-m} + f_{n-m+1}\ \alpha + \cdots + f_{n-1}\ \alpha^{m-1} + f_n\ \alpha^m .$$

However, since α is a root of $g(x)$,

$$\alpha^m = -g_0 - g_1\ \alpha - \cdots - g_{m-1}\ \alpha^{m-1} .$$

Thus the contents of the register with feedback temporarily disabled is equivalent to

$$(f_{n-m} - f_n\ g_0) + (f_{n-m+1} - f_n\ g_1)\ \alpha + \cdots + (f_{n-1} - f_n\ g_{m-1})\ \alpha^{m-1}$$

which is just the contents of the register if feedback were allowed. In other words, the feedback replaces α^m by its equivalent in terms of $1, \ldots, \alpha^{m-1}$. Continuing this reasoning for $n + 1$ shifts it is observed that the coefficients of $f(\alpha) = f_0 + f_1 \alpha + \ldots + f_n \alpha_1^n = r(\alpha)$ remain in the register.

If the input, $f(x)$, to the division circuit is zero and the register contents correspond to $r(\alpha) = r_0 + r_1 \alpha + \ldots + r_{m-1} \alpha^{m-1}$ initially, then after one shift the register contents correspond to $\alpha\, r(\alpha)$ or equivalently $x\, r(x) \bmod g(x)$. Therefore, this circuit can be used to count in GF(p^m). If a 1 is initially placed in the lowest order storage element, then the shift register contents become 1, α, α^2, . . . as it is shifted.

In some cases it is necessary to calculate $\alpha^k f(x)$ where k is a positive integer. The division circuit can be modified as shown in Figure 2 to perform the premultiplication by α^k. The operation of the circuit can be described as follows:

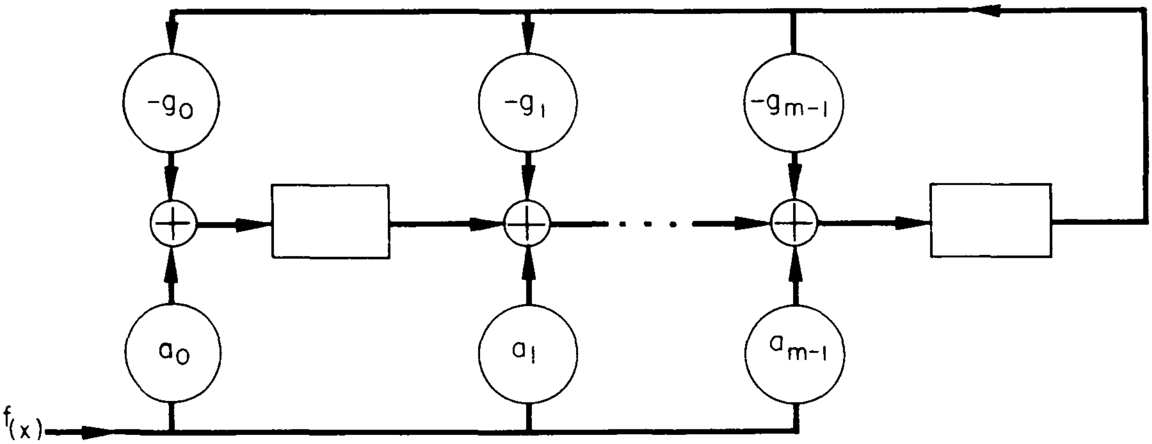

Figure 2–Circuit for Calculating $\alpha^k f(x)$

If $x^k \bmod g(x) = a(x)$ then $\alpha^k = a(\alpha) = a_0 + a_1 \alpha + \ldots + a_{m-1} x^{m-1}$. After the first shift the register contents are $f_n \alpha^k$. After the second shift the contents are $\alpha^{k+1} f_n + \alpha^k f_{n-1}$. Continuing this reasoning it is clear that after $n + 1$ shifts the register contains $\alpha^k f(\alpha)$ or equivalently $x^k f(x) \bmod g(x)$.

Example 17

Let $g(x) = x^3 + x + 1$ as in Example 14 and let $k = m = 3$. Then $\alpha^3 = \alpha + 1$ and the circuit for finding $\alpha^3 f(\alpha)$ is shown in Fig. 3.

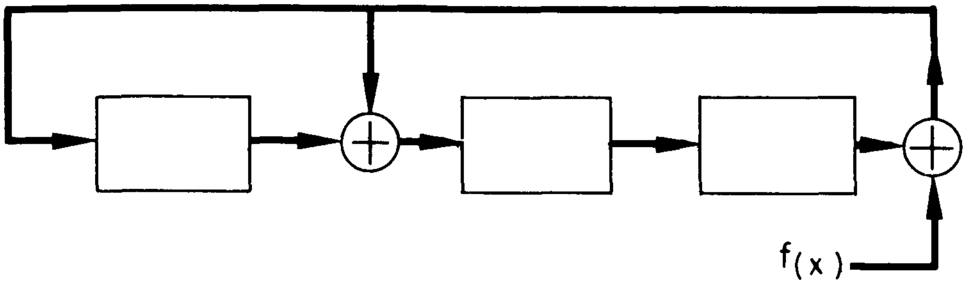

Figure 3–Circuit for Finding α^3 f (x)

Example 18

Let $g(x) = x^3 + x + 1$ again and $k = 5$. The circuit for finding $\alpha^5 f(\alpha)$ is shown in Fig. 4.

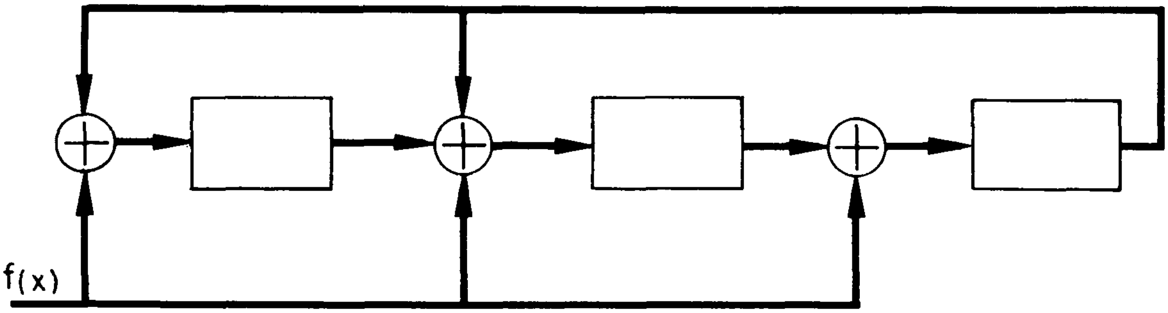

Figure 4–Circuit for Finding α^5 f (x)

IV. CYCLIC BLOCK CODES

BCH codes belong to the class of algebraic codes known as cyclic codes. In this section cyclic codes are defined and some of their properties explained.

Definition 18: Least common multiple

Given the set of integers $e_1, e_2, \ldots, e_m$, the least common multiple of these integers is the smallest number divisible by each of them.

Given a set of polynomials $e_1(x), \ldots, e_m(x)$, their least common multiple is the polynomial of lowest order divisible by all of them.

Let $g(x)$ be a polynomial of order r over $GF(p)$ with roots $\alpha_1, \alpha_2, \ldots, \alpha_r$ that are elements of $GF(p^m)$. Let the orders of the roots be $e_1, \ldots, e_r$ and let n be the least common multiple of the orders.

Then

$$n = q_1 e_1 = q_2 e_2 = \cdots = q_r e_r$$

and

$$\alpha_i^n = \alpha_2^n = \cdots = \alpha_r^n = 1$$

since

$$\alpha_i^n = \alpha_i^{q_i e_i} = (\alpha_i^{e_i})^{q_i} = 1^{q_i} = 1.$$

Therefore each root of $g(x)$ is a root of $x^n - 1$ so that $g(x)$ divides $x^n - 1$ and n is the smallest integer such that $x^n - 1$ is divisible by $g(x)$.

Consider the set of all $n - 1$ order polynomials of the form

$$f(x) = q(x)\, g(x) \bmod (x^n - 1)$$

where $g(x)$ is an arbitrary polynomial over $GF(p)$ and n is the samllest integer such that $x^n - 1$ is divisible by $g(x)$. It is shown in the following two theorems that each of these polynomials is divisible by $g(x)$ and that the set is closed under addition thus forming a linear vector subspace.

Theorem III

Let n be the smallest integer such that $x^n - 1$ is divisible by $g(x)$. Then $q(x)\, g(x) \bmod (x^n - 1)$ is also divisible by $g(x)$.

Proof:

According to the Euclidean division algorithm

$$q(x)\, g(x) = h(x)\, (x^n - 1) + r(x)$$

where h (x) is the quotient and q (x) g(x) mod $(x^n - 1) = r$ (x) by definition. Clearly, g (x) divides the left hand side of this equation. g (x) divides $x^n - 1$ and therefore it must also divide r(x) if the right hand side is to be divisible by g(x).

Q.E.D.

Theorem IV

Let $f_1(x) = q_1(x) g(x) \bmod (x^n - 1)$ and $f_2(x) = q_2(x) g(x) \bmod (x^n - 1)$. Then $f_1(x) + f_2(x)$ is divisible by g(x).

Proof:

According to the previous theorem both $f_1(x)$ and $f_2(x)$ are divisible by g(x). Therefore the sum is divisible by g(x).

Q.E.D.

The set of polynomials q(x) g(x) mod $(x^n - 1)$ will be taken as the code vectors. From the previous two theorems it is clear that these vectors form a subspace of an n dimension vector space and that each code vector is a multiple of g(x). g(x) is called the generator polynomial for the code and n is called the code length.

If r is the order of g (x) it can be shown [18] that the code vectors form an n-r dimensional subspace. That is, n-r components can be arbitrarily chosen as information symbols and the remaining r symbols must be check symbols. In most cases, it is convenient to chose the first n - r components, i.e., the coefficients of x^{n-1}, x^{n-2}, . . . , x^2, as information symbols. Codes of this type are called systematic codes. Encoding can be performed as follows:

Let the information correspond to the polynomial $I(x) = i_{n-1} x^{n-1} + \ldots + i_r x^r$. Then according to the Euclidean division algorithm

$$I(x) = q(x) g(x) + c(x)$$

where $c(x) = c_{r-1} x^{r-1} + \ldots + c_0$ is the remainder when I(x) is divided by g(x). Therefore

$$q(x) g(x) = I(x) - c(x) = i_{n-1} x^{n-1} + \cdots + i_r x^r - c_{r-1} x^{r-1} + \cdots - c_0$$

is a code word with the information symbols appearing first. The encoding can be performed by dividing I (x) by g (x) to find c (x) using the circuits discussed in the previous section. The reader desiring a more complete discussion and alternate circuits should consult Peterson [18].

Given any code word

$$f(x) = f_{n-1} x^{n-1} + f_{n-2} x^{n-2} + \cdots + f_0,$$

then

$$x\, f(x) = f_{n-1} x^{n} + f_{n-2} x^{n-1} + \cdots + f_0 x$$

must also be a code word since it is still a multiple of g (x). Since $x^n \bmod (x^n - 1) = 1$, $x\, f(x) \bmod (x^n - 1) = f_{n-2} x^{n-1} + \ldots + f_0 x + f_{n-1}$. If f (x) is represented in the vector form $f(x) = (f_0, f_1, \ldots f_{n-1})$. Then $x\, f(x) = (f_{n-1}, f_0, f_1, \ldots, f_{n-2})$ is just a cyclic shift of the vector f (x). This is why these are called cyclic codes.

V. BOSE-CHAUDHURI-HOCQUENGHEM CODES

A. Definition

BCH codes are cyclic codes with symbols in GF(p) and can most easily be described in terms of their generator polynomials. Let α be an element of an extension field GF(p^m), for example, a root of a primitive polynomial of order m over GF(p) and let m_0 be an arbitrary integer. Then the generator polynomial g (x) is the polynomial of smallest degree that has

$$\alpha^{m_0}, \alpha^{m_0+1}, \cdots, \alpha^{m_0+d-2}$$

as roots. Since each code vector f (x) is a multiple of g (x), these must also be roots of f (x). The cases where $m_0 = 0$ or 1 are most frequently used. If $m_0 = 0$ then 1 is a root of f (x) so that

$$f(1) = \sum_{i=0}^{n-1} f_i = 0$$

which corresponds to a simple parity check. The code length n is the smallest integer such that $g(x)$ divides $x^n - 1$. It can be shown [18] that n is the order of α for $d > 2$, for example, $n = p^m - 1$ if α is primitive. The code vectors are the set of all polynomials of the form $q(x)\ g(x) \bmod (x^n - 1)$.

Example 19.

Let α be a root of the primitive polynomial $x^3 + x + 1$ as in Example 14. Then the code length is $n = 2^3 - 1 = 7$. Let $m_0 = 1$ and $d = 3$. Then α and α^2 must be roots of $g(x)$. According to Corollary 1, if α is a root of polynomial then α^2 is also a root. Therefore $g(x) = x^3 + x + 1$ and the code vectors contain four information symbols and three check symbols. For example, let the information symbols be $x^6 + x^3$, then

$$\begin{array}{r|l} & x^3 + x \\ \cline{2-2} x^3 + x + 1 & x^6 + 0 + 0 + x^3 \\ & x^6 + x^4 + x^3 \\ \cline{2-2} & \quad x^4 \\ & \quad x^4 + x^2 + x \\ \cline{2-2} & \qquad\quad x^2 + x \end{array}$$

Therefore the check symbols are $x^2 + x$ and the code vector becomes

$$f(x) = x + x^2 + x^3 + x^6 = (0, 1, 1, 1, 0, 0, 1)$$

As an exercise the reader might show that for $d > 3$ the code becomes degenerate and consists only of check symbols.

Stenbit [22] has calculated the generator polynomials for all nontrivial BCH codes up to length 255 with $m_0 = 1$ and α a primitive element. Peterson [18] contains a table of irreducible polynomials over GF (2) and minimum polynomials for powers of primitive elements which can be used for calculating the generator polynomials for additional codes.

B. Distance Structure

It can be shown [18] that the minimum Hamming distance of the BCH codes is at least d. Therefore to insure that the codes are capable of correcting t errors d must be chosen to be at least $2t + 1$. These codes will then detect $2t = d - 1$ or less errors.

For binary codes when $m_0 = 1$ and $d = 2t + 1$,

$$\alpha, \cdots, \alpha^{2t}$$

must be roots of $g(x)$. Since every even power of α is a root of the minimum polynomial of some odd power of α, $g(x)$ is the least common multiple of $m_1(x)$, $m_3(x), \ldots, m_{2t-1}(x)$. If α is an element of $GF(2^m)$, then each minimum polynomial has order m or less and $g(x)$ is the product of at most t polynomials of order m so that the order of $g(x)$ is at most mt. Therefore if α is a primitive root, $g(x)$ generates a code of length $2^m - 1$ with at most mt check digits. The generator polynomials for codes with various t and n are given in Stenbit [22].

If $g(x)$ has order $n - k$, where k is the number of information symbols, the BCH codes will also detect all bursts of length $n - k$ or less. A burst of length L is an error pattern of the form

$$x^r (b_0 + b_1 x + \cdots + b_{L-1} x^{L-1}) = x^r B(x)$$

where b_i's are elements of $GF(p)$ and $b_0 \neq 0$, $b_{L-1} \neq 0$. Since x^r is not divisible by $g(x)$ because no root of $g(x)$ raised to the r^{th} power is zero and $B(x)$ is not divisible by $g(x)$ if $L \leq n - k$, the burst cannot be a code vector.

It should also be observed that if $m_0 = 1$ and $d = 3$ with α a primitive root, the BCH codes are equivalent to the well known Hamming codes. In this case α and α^2 must be roots of $g(x)$ so that $g(x)$ is just the primitive polynomial with α as its root as in Example 19.

C. Error Detection

Since each transmitted code vector $f(x)$ is a multiple of $g(x)$ it is only necessary to divide the received vector by $g(x)$ to check for errors. If a detectable error has occurred the remainder will not be zero. If no error or an undetectable error has occurred the remainder will be zero. The division can be performed by shifting $f(x)$ into the division circuit shown in Fig. 1.

D. Error Correction

All known error correction schemes are based on Newton's identities relating the elementary symmetric functions to the power sum symmetric functions. Since these relations are not clearly explained in the literature on BCH codes, they will be derived here using the method of Bocher [3]. With this background the reader should be able to understand the literature [9, 11, 15, 17, 18, 24, 25] without serious difficulty and therefore these decoding methods will not be discussed here.

Definition: Elementary Symmetric Functions

Given a set of numbers $x_1, \ldots, x_n$, the polynomial

$$\begin{aligned}\sigma(x) &= (x - x_1)(x - x_2) \cdots (x - x_n) \\ &= x^n - \sigma_1 x^{n-1} + \sigma_2 x^{n-2} + \cdots + (-1)^n \sigma_n\end{aligned}$$

is formed. The coefficients $\sigma_1, \ldots, \sigma_n$ are defined as the elementary symmetric functions of $x_1, \ldots, x_n$. If the σ_i's are calculated, it is found that

$$\sigma_1 = \sum_{i=1}^{n} x_i$$

$$\sigma_2 = \sum x_i x_j \text{ for all different combinations of } i \text{ and } j$$

$$\sigma_3 = \sum x_i x_j x_k \text{ for all different combinations of } i, j, k.$$

$$\vdots$$

$$\sigma_n = x_1 x_2 \cdots x_n$$

If any two numbers x_i and x_j are interchanged the values of the symmetric functions remain the same. Therefore the functions $\sigma_1, \ldots, \sigma_n$, are said to be symmetric with respect to the variables $x_1, \ldots, x_n$.

Definition: Power Sum Symmetric Functions

Given the set of numbers $x_1, \ldots, x_n$, the k^{th} power sum symmetric function of these numbers is defined to be

$$S_k = \sum_{i=1}^{n} x_i^k$$

The elementary symmetric functions and power sum symmetric functions are related by a set of linear equations known as Newton's identities. Using the factored form of $\sigma(x)$

$$\frac{d\sigma}{dx} = \frac{\sigma}{x - x_1} + \frac{\sigma}{x - x_2} + \cdots + \frac{\sigma}{x - x_n}$$

Since

$$\sigma(x_i) = 0$$

$$\sigma(x) = (x^n - x_i^n) - \sigma_1 (x^{n-1} - x_i^{n-1}) + \cdots$$

and

$$\frac{\sigma(x)}{x - x_i} = x^{n-1} + (x_i - \sigma_1) x^{n-2} + (x_i^2 - \sigma_1 x_i + \sigma_2) x^{n-3} + \cdots$$

and

$$\frac{d\sigma}{dx} = n x^{n-1} + (S_1 - n\sigma_1) x^{n-2} + (S_2 - \sigma_1 S_1 + n\sigma_2) x^{n-3} + \cdots$$

From the unfactored form

$$\frac{d\sigma}{dx} = n x^{n-1} (n-1) \sigma_1 x^{n-2} + (n-2) \sigma_2 x^{n-3} + \cdots$$

Equating coefficients of like powers of x in these two expressions gives

$$S_1 - n\sigma_1 = -(n-1)\sigma_1$$

$$S_2 - \sigma_1 S_1 + n\sigma_2 = (n-2)\sigma_2$$

$$\vdots$$

$$S_{n-1} - \sigma_1 s_{n-2} + \sigma_2 S_{n-3} - \cdots + (-1)^{n-1} n\sigma_{n-1} = (-1)^{n-1} \sigma_{n-1}$$

or

$$S_1 - \sigma_1 = 0$$

$$S_2 - \sigma_1 s_1 + 2\sigma_2 = 0$$

$$\cdot$$
$$\cdot$$
$$\cdot$$

$$S_{n-1} - \sigma_1 s_{n-2} + \sigma_2 s_{n-3} - + \cdots + (-1)^{n-1} (n-1)\sigma_{n-1} = 0$$

VI. SHORTENED BCH CODES

Codes of length different from $p^m - 1$ may be desired owing to equipment or format specifications. These are easily obtained by simply making some of the initial information symbols 0 in an unshortened BCH code and not transmitting these symbols. Since the shortened words are still code vectors, the minimum Hamming distance remains unchanged so that the error correction and detection capabilities are unchanged. Encoding and decoding procedures for the natural length codes also apply to the shortened codes.

An alternative method for obtaining codes of different length is to chose α as a nonprimitive element of GF (p^m). A table of some binary BCH codes generated by nonprimitive elements can be found in Peterson (18).

REFERENCES

1. Bartee, T. C. and D. I. Schneider, "An Electronic Decoder for Bose-Chaudhuri-Hocquenghem Error Correcting Codes," IRE Transactions on Information Theory, vol. IT-8, no. 5 pp. S 17-24, September 1962.

2. Berlekamp, E. R., "On Decoding Binary Bose-Chaudhuri-Hocquenghem Codes," IEEE Transactions on Information Theory," vol. IT-11, no. 4, pp. 577-80, October 1965.

3. Bochner, M., *Introduction to Higher Algebra*, Macmillan Co., New York, 1907, Chapter XVIII.

4. Bose, R. C. and R. R. Kuebler, "On the Construction of a Class of Error Correcting Binary Signating Codes," Technical Report, University of North Carolina, Chapel Hill, N. C., May 1958.

5. Bose, R. C. and S. S. Shrikhande, "A Note on a Result in the Theory of Code Construction," *Information and Control*, 2, pp. 183-194, 1959.

6. Bose, R. C. and P. K. Ray Chaudhuri, "On a Class of Error Correcting Binary Group Codes," *Information and Control*, 3, pp. 68-79, 1960.

7. Bose, R. C. and P. K. Ray Chaudhuri, "Further Results on Error Correcting Binary Group Codes," *Information and Control*, 3, pp. 279-290, 1960.

8. Chien, R. T. and D. T. Tang, "On Detecting Errors After Correction," Proceedings of the IEEE, vol. 62, p. 974, August 1964.

9. Chien, R. T., "Cyclic Decoding Procedures for Bose-Chaudhuri-Hocquenghem Codes," IEEE Transactions on Information Theory, vol. IT-10, pp. 357-363, October 1964.

10. Chien, R. T. and V. Lum, "On Golay's Perfect Codes and Step-by-Step Decoding," IEEE Transactions on Information Theory, vol. IT-2, pp. 403-404, July 1966.

11. Forney, G. D., "On Decoding BCH Codes," IEEE Transactions on Information Theory, vol. IT-11, pp. 549-557, October 1965.

12. Forney, G. P., "Generalized Minimum Distance Decoding," IEEE Transactions on Information Theory, vol. IT-12, pp. 125-131, April 1966.

13. Gross, A. J., "Augmented Bose-Chaudhuri Codes which Correct Single Bursts of Errors," IEEE Transactions on Information Theory, vol. IT-9, no. 2, p. 121, April 1963.

14. Hocquenghem, A., "Codes Correcteurs d'erreurs," Chiffres, vol. 2, pp. 147-156, September 1959.

15. Massey, J. L., "Step-by-Step Decoding of the Bose-Chaudhuri-Hocquenghem Codes," IEEE Transactions on Information Theory, vol. IT-11, pp. 580-585, October 1965.

16. Mattson, H. F. and G. Solomon, "A New Treatment of Bose-Chaudhuri Codes," J. SIAM, vol. 9, pp. 654-670, December 1961.

17. Peterson, W. W., "Encoding and Error-Correction Procedures for BCH Codes," IRE Transactions on Information Theory, vol. IT-6, pp. 459-470, September 1960.

18. Peterson, W. W., Error Correcting Codes, M.I.T. Press and John Wiley & Sons, Inc., 1961.

19. Peterson, W. W. and J. L. Massey, "Coding Theory, URSI Report of Progress in Information Theory in the United States, 1960-1963," IEEE Transactions on Information Theory, vol. IT-9, pp. 223-229, October 1963.

20. Polinghorn, F., "Decoding of Double and Triple Error Correcting Bose-Chaudhuri Codes," IEEE Transactions on Information Theory, vol. IT-I2, no. 4, pp. 480-82, October 1966.

21. Reed, I. S. and G. Solomon, "Polynomial Codes Over Certain Finite Fields," J. SIAM, vol. 8, pp. 300-304, 1960.

22. Stenbit, J. P., "Table of Generators for Bose-Chaudhuri Codes," IEEE Transactions on Information Theory, vol. IT-10, no. 4, pp. 390-91, October 1964.

23. Szwaja, Z., "On Step-by-Step Decoding of BCH Binary Codes," IEEE Transactions on Information Theory, vol. IT-13, no. 2, pp. 350-51, April 1967.

24. Wozencraft, J. M. and I. M. Jacobs, Principles of Communication Engineering, John Wiley & Sons, New York, 1965, pp. 441-443.

25. Zieler, N., "A Class of Cyclic Linear Error-Correcting Codes in p^m Symbols," MIT Lincoln Labs Group Report 55-19, Lexington, Mass., January 1960.

26. Zieler, N. and D. Gorenstein, "A Class of Error Correcting Codes in p^m Symbols," J. SIAM, vol. 9, no. 2, pp. 207-214, June 1961.

APPENDIX 1
Example of Triple Error Correcting (15, 5) BCH Code

Let α be a root of the primitive polynomial $m_1(x) = x^4 + x + 1$. Then any element in $GF(2^4)$ can be represented in the form

$$c_0 + c_1 \alpha + c_2 \alpha^2 + c_3 \alpha^3.$$

a table for the representation is shown below.

	c_0	c_1	c_2	c_3
1	1	0	0	0
α	0	1	0	0
α^2	0	0	1	0
α^3	0	0	0	1
α^4	1	1	0	0
α^5	0	1	1	0
α^6	0	0	1	1
α^7	1	1	0	1
α^8	1	0	1	0
α^9	0	1	0	1
α^{10}	1	1	1	0
α^{11}	0	1	1	1
α^{12}	1	1	1	1
α^{13}	1	0	1	1
α^{14}	1	0	0	1
α^{15}	1	0	0	0

Representation of $GF(2^4)$.

According to Theorem II α, α^2, α^4 and α^8 are all roots of the same polynomial. Therefore the minimum polynomial for each of these elements is $m_1(x) = x^4 + x + 1$.

α^3, α^6, α^{12}, and $\alpha^{24} = \alpha^9$ all must be roots of the minimum polynomial $m_3(x)$ for α^3. If

$$m_3(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3 + x^4$$

then

$$m_3(\alpha^3) = c_0 + c_1 \alpha^3 + c_2 \alpha^6 + c_3 \alpha^9 + \alpha^{12} = 0$$

or according to the table

$$c_0 \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + c_1 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} + c_2 \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} + c_3 \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} .$$

The solution of this set of four equations is

$$c_0 = 1, \; c_1 = 1, \; c_2 = 1, \text{ and } c_3 = 1$$

so

$$m_3(x) = 1 + x + x^2 + x^3 + x^4 .$$

α^5 and α^{10} are the roots of

$$m_5(x) = (x - \alpha^5)(x - \alpha^{10}) = x^2 + (\alpha^5 + \alpha^{10})x + \alpha^{15}$$

$$= x^2 + x + 1$$

Let $m_0 = 1$ and $d = 7$, then $\alpha, \alpha^2, \ldots, \alpha^6$ must be roots of $g(x)$. Therefore

$$g(x) = m_1(x)\, m_3(x)\, m_5(x)$$

$$= x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1$$

The minimum polynomials could have been found directly from the tables in Peterson [18] and g (x) from Stenbit [22].

The circuit shown in Figure 5 can be used for encoding. The five information digits are simultaneously shifted into the register and transmitted over the channel. Since this circuit automatically premultiplies its input by x^{10}, the check bits I (x) mod g (x) remain in the register after the information bits have been shifted into it. The feedback is then disabled and the check bits are transmitted over the channel.

An identical circuit can be used for error detection except that the entire 15 bit received vector r (x) is shifted into the register. The circuit calculates x^{10} r (x) mod g (x). For error correction, the input must be added into the first stage of the register rather than at the end so that r (x) mod g(x) is calculated.

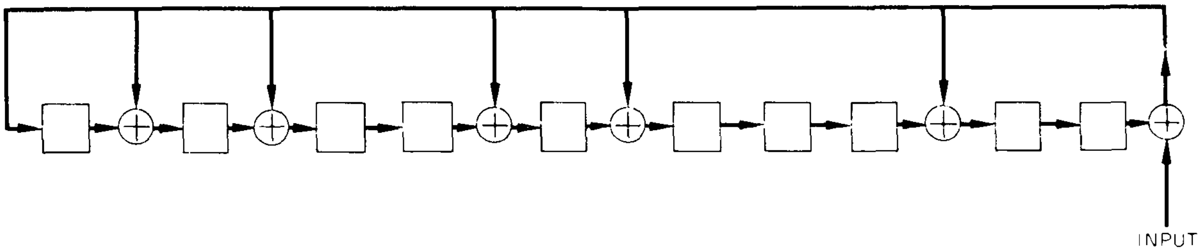

Figure 5–Encoding Circuit for (15, 5) BCH Codes