the (1, 1) indicates not orbit 5.1, which it would if $\varphi(s)$ had been equal to (2, 0, 1), but rather orbit 5.2. We have, symbolically, (2, 1, 1) = (1, 0, 2) which includes (3, 0, 0) = (0, 1, 3), which is certainly possible. Naturally, the decoder could not know at this point that $\varphi(s) = (1, 1, 1)$ rather than (2, 0, 1), although it could avoid the difficulty by searching for $h_i(s) = (1, 9)$ first. However, the decoder performs correctly without the preliminary search, for it would not find j with $h_j\ (s + e_i) = (-1, -9)$. Indeed, in order to do so we should have to have (3, 1, 1) = (2, 0, 2) = (0, 0, 1), and the last equation is equivalent to (2, 0, 0) = (0, 0, 3) or (2, 0, 0) = (0, 0, 1), which are impossible. The other facts used in the procedure may be established in a similar way.

Bibliography

[1] W. Burnside, "Theory of Groups of Finite Order," Dover Publications, Inc., New York, N. Y.; 1955.

[2] M. J. E. Golay, "Notes on digital coding," Proc. IRE, vol. 37, p. 657; June, 1949.

[3] M. Hall, "Projective Planes and Related Topics," California Institute of Technology, Pasadena; April, 1954.

[4] L. J. Paige, "A note on the Mathieu groups," *Can. J. Math.*, vol. 9, pp. 15–18; January, 1956.

[5] E. Prange, "Cyclic Error-Correcting Codes in Two Symbols," AFCRC-TN-57-103, ASTIA Document No. AD 133749; September, 1957.

[6] E. Prange, "Some Cyclic Error-Correcting Codes with Simple Decoding Algorithms," AFCRC-TN-58-156, ASTIA Document No. AD 152386; April, 1958.

[7] D. Slepian, "A class of binary signaling alphabets," *Bell Sys. Tech. J.*, vol. 35, pp. 203–234; January, 1956.

[8] E. Prange, "The Use of Coset Equivalence in the Analysis and Decoding of Group Codes," AFCRC-TR-59-164; June, 1959.

Encoding and Error-Correction Procedures for the Bose-Chaudhuri Codes*

W. W. PETERSON†, MEMBER, IRE

Summary—**Bose and Ray-Chaudhuri have recently described a class of binary codes which for arbitrary m and t are t-error correcting and have length $2^m - 1$ of which no more than mt digits are redundancy. This paper describes a simple error-correction procedure for these codes. Their cyclic structure is demonstrated and methods of exploiting it to implement the coding and correction procedure using shift registers are outlined. Closer bounds on the number of redundancy digits are derived.**

Introduction

BOSE and Chaudhuri[1] have recently discovered a new class of codes with some remarkable properties. For any positive integers m and t, there is a code in this class that consists of blocks of length $2^m - 1$, that corrects t errors, and that requires no more than mt parity check digits. Thus, the codes cover a wide range in rate and error-correcting ability, unlike most other known classes of codes.[2] These codes are a generalization of the Hamming codes;[3] the case $t = 1$ gives the Hamming code in each case.

In this paper two important properties of these codes are described. First, a method for error correction is described which is a generalization of the simple error-correction procedure that can be used with Hamming codes. The procedure requires a number of operations which increases only as a small power of the length of the codes.

Second, it is shown that these are cyclic codes[4] and,

* Received by the PGIT, December 6, 1959. Part of this work was supported by the U. S. Army Signal Corps, the U. S. Air Force Office of Scientific Research, Air Research and Development Command, and the U. S. Navy Office of Naval Research at the Research Laboratory of Electronics, Mass. Inst. Tech., Cambridge, Mass.; and part of the work was done at the IBM Research Lab., Yorktown, N. Y.

† On leave from the University of Florida, Gainesville. Presently at the Dept. of Elec. Engrg. and Res. Lab. of Electronics, Mass. Inst. Tech., Cambridge, Mass.

[1] R. C. Bose and D. K. Ray-Chaudhuri, "On a class of error-correcting binary group codes," to be published in *Information and Control.*

[2] The only others of which I am aware are I. S. Reed, "A class of multiple-error-correcting codes and decoding scheme," IRE Trans. on Information Theory, vol. IT-4, pp. 38–49, September, 1954; P. Elias, "Error free coding," IRE Trans. on Information Theory, vol. IT-4, pp. 29–37, September, 1954; and I. S. Reed and G. Solomon, "Polynomial code," to be published in *J. Soc. Ind. Appl. Math.*

[3] R. W. Hamming, "Error detecting and error correcting codes," *Bell Sys. Tech. J.*, vol. 29, pp. 147–160; April, 1950.

[4] E. Prange, "Some Cyclic Error-Correcting Codes with Simple Decoding Algorithms," Air Force Cambridge Research Center, Bedford, Mass., Tech. Note AFCRC-TN-58-156, April, 1958; "Cyclic Error-Correcting Codes in Two Symbols," Air Force Cambridge Research Center, Bedford, Mass., Tech. Note ARCRC-TN-57-103, September, 1957; "The Use of Coset Equivalence in the Analysis and Decoding of Group Codes," Air Force Cambridge Research Center, Bedford, Mass., Tech. Rept. AFCRC-TR-59-164, June, 1959.

therefore, the encoding can be accomplished very efficiently with a shift register. The theory of the cyclic structure also provides a closer bound on the number of parity checks required to correct a given number of errors.

Construction of the Bose-Chaudhuri Codes

Given an irreducible polynomial $p(X)$ of degree m with 1 and 0 as coefficients, a representation of the Galois Field with 2^m elements $GF(2^m)$ can be formed. It consists of all polynomials of degree $m - 1$ or less. They can be added (modulo 2) term by term in the ordinary way. The rule for multiplication is to multiply in the ordinary way, reducing the answer modulo 2 and modulo $p(X)$ to a polynomial of degree $m - 1$ or less. (That is, consider $p(X) = 0$, and use this equation to eliminate terms of power greater than $m - 1$.) It can be shown then that certain of these polynomials, called primitive elements, have the property that the first $2^m - 1$ powers of such an element are exactly all the $2^m - 1$ nonzero field elements. Also, every nonzero field element is a root of the equation

$$X^{2^m-1} = 1$$

and conversely. Thus if α is any element of the field, $\alpha^{-1} = \alpha^{2^m-2}$.

The field elements can also be thought of as vectors whose components are the coefficients of the polynomials. The sum of two vectors corresponds to the sum of the corresponding polynomials.

The Bose-Chaudhuri codes are described by giving the matrix of parity check rules, which is the matrix

$$M = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \alpha & \alpha^3 & \cdots & \alpha^{2t-1} \\ \alpha^2 & (\alpha^3)^2 & \cdots & (\alpha^{2t-1})^2 \\ \vdots & \vdots & \vdots\;\vdots\;\vdots & \vdots \\ \alpha^{2^m-2} & (\alpha^3)^{2^m-2} & \cdots & (\alpha^{2t-1})^{2^m-2} \end{bmatrix} \tag{1}$$

where α is a primitive element of the field.

This is a $2^m - 1 \times t$ matrix of $GF(2^m)$ elements, but thinking of each field element as a vector of m binary digits, this is a $2^m - 1 \times mt$ matrix of binary digits. A vector of $2^m - 1$ binary digits is considered a code word if it satisfies the parity check described by each column; *i.e.*, if the product of this vector with the matrix is zero. In other words the set of all code words is the (left) null space of this matrix.

The code that Bose and Ray-Chaudhuri use as an example will be used to illustrate the ideas discussed in this paper. Let α denote a root of the equation $X^4 = X + 1$. This happens to be a primitive element of the field. Then the 15 nonzero field elements are given in Table I.

Taking $t = 3$, the following matrix of parity check rules results:

TABLE I
Representation of $GF(2^4)$

α^0	=	1	=	(1 0 0 0)
α^1	=	α	=	(0 1 0 0)
α^2	=	α^2	=	(0 0 1 0)
α^3	=	α^3	=	(0 0 0 1)
α^4	=	$1 + \alpha$	=	(1 1 0 0)
α^5	=	$\alpha + \alpha^2$	=	(0 1 1 0)
α^6	=	$\alpha^2 + \alpha^3$	=	(0 0 1 1)
α^7	=	$1 + \alpha + \alpha^3$	=	(1 1 0 1)
α^8	=	$1 + \alpha^2$	=	(1 0 1 0)
α^9	=	$\alpha + \alpha^3$	=	(0 1 0 1)
α^{10}	=	$1 + \alpha + \alpha^2$	=	(1 1 1 0)
α^{11}	=	$\alpha + \alpha^2 + \alpha^3$	=	(0 1 1 1)
α^{12}	=	$1 + \alpha + \alpha^2 + \alpha^3$	=	(1 1 1 1)
α^{13}	=	$1 + \alpha^2 + \alpha^3$	=	(1 0 1 1)
α^{14}	=	$1 + \alpha^3$	=	(1 0 0 1)
α^{15}	=	$1 = \alpha^0$		

$$M = \left[\begin{array}{ccc} 1\,0\,0\,0 & 1\,0\,0\,0 & 1\,0\,0\,0 \\ 0\,1\,0\,0 & 0\,0\,0\,1 & 0\,1\,1\,0 \\ 0\,0\,1\,0 & 0\,0\,1\,1 & 1\,1\,1\,0 \\ 0\,0\,0\,1 & 0\,1\,0\,1 & 1\,0\,0\,0 \\ 1\,1\,0\,0 & 1\,1\,1\,1 & 0\,1\,1\,0 \\ 0\,1\,1\,0 & 1\,0\,0\,0 & 1\,1\,1\,0 \\ 0\,0\,1\,1 & 0\,0\,0\,1 & 1\,0\,0\,0 \\ 1\,1\,0\,1 & 0\,0\,1\,1 & 0\,1\,1\,0 \\ 1\,0\,1\,0 & 0\,1\,0\,1 & 1\,1\,1\,0 \\ 0\,1\,0\,1 & 1\,1\,1\,1 & 1\,0\,0\,0 \\ 1\,1\,1\,0 & 1\,0\,0\,0 & 0\,1\,1\,0 \\ 0\,1\,1\,1 & 0\,0\,0\,1 & 1\,1\,1\,0 \\ 1\,1\,1\,1 & 0\,0\,1\,1 & 1\,0\,0\,0 \\ 1\,0\,1\,1 & 0\,1\,0\,1 & 0\,1\,1\,0 \\ 1\,0\,0\,1 & 1\,1\,1\,1 & 1\,1\,1\,0 \end{array}\right] \tag{2}$$

Of these twelve columns, the last one is trivial and the next to last is a duplicate; these two can be dropped. The rest are independent, and the result is a code with fifteen digit code words of which ten are parity checks and five are information places. The code corrects all triple errors.

An Error-Correction Procedure

Consider the result of multiplying a vector $(r_0, r_1, r_2, \cdots, r_{n-1})$ of $n = 2^m - 1$ components by the matrix M in (1). The result is a vector of t Galois field elements. The first component is

$$r_0 + r_1\alpha + r_2\alpha^2 + \cdots + r_{n-1}\alpha^{n-1} = r(\alpha)$$

where

$$r(X) = r_0 + r_1X + \cdots + r_{n-1}X^{n-1}$$

is the polynomial which corresponds naturally to the given vector. (In what follows no distinction will be made between a vector and the corresponding polynomial.) The other components are clearly $r(\alpha^3), r(\alpha^5), \cdots, r(\alpha^{2t-1})$.

In these terms an equivalent definition of the Bose-Chandhuri codes can be given. A vector is a code word if it is in the left null space of M, *i.e.*, if the parity checks $r(\alpha)$, $r(\alpha^3)$, $r(\alpha^5)$, $\cdots$, $r(\alpha^{2t-1}0)$ are zero. This can be restated as follows:

Definition: A polynomial $s(X)$ is a code vector for a t-error correcting Bose-Chaudhuri code if, and only if, $\alpha, \alpha^3, \cdots, \alpha^{2t-1}$ are roots of $s(X)$.

The first step in devising a decoding method is to characterize the information contained in the parity check calculation for a received vector which may contain errors. Let $e = (e_0, e_1, \cdots, e_{n-1})$ be the vector of errors, *i.e.*, if the errors occur in the positions $i_1, i_2, \cdots, i_v$, then

$$e_i = 1 \text{ for } i = i_1, i_2, \cdots, i_v$$

$$e_i = 0 \text{ otherwise.}$$

There is a one to one correspondence between the elements of the error vector and the elements of $GF(2^m)$ which constitute the first column of the parity check matrix M given by (1), e_i corresponding to the element a^i occurring in the i-th position in the first column of M. The elements $X_1, X_2, \cdots, X_v$ of $GF(2^m)$ which correspond in this way to $e_{i_1}, e_{i_2}, \cdots, e_{i_v}$ may be called the error position numbers. Thus $X_j = a^{i_j}$ $(j = 1, 2, \cdots, v)$.

Lemma 1: If a received vector r has errors in digits numbered $X_1, X_2, \cdots, X_v$, then the parity check vector $r \times M$ is of the form $(S_1, S_3, S_5, \cdots, S_{2t-1})$ where

$$S_j = \sum_{i=1}^{v} X_i^j. \tag{3}$$

Proof: Assume that the vector s was transmitted, and $r = s + e$ received, where e has ones in the positions $i_1, i_2, \cdots, i_v$ and zeros in all other positions. In terms of corresponding polynomials,

$$r(X) = s(X) + e(X)$$

and the result of the parity check calculation is

$$[r(\alpha), r(\alpha^3), \cdots, r(\alpha^{2t-1})].$$

But $s(\alpha) = s(\alpha^3) = \cdots = s(\alpha^2{}_t{}^{-1}) = 0$, so that $r(\alpha) = s(\alpha) + e(\alpha) = e(\alpha)$, $r(\alpha^3) = e(\alpha^3)$, etc. Thus, the result of the parity check calculation is $[e(\alpha), e(\alpha^3), \cdots, e(\alpha^{2t-1})]$. But

$$e(\alpha^j) = e_0 + e_1\alpha^j + e_2\alpha^{2j} + \cdots + e_{n-1}\alpha^{(n-1)j}$$

$$= \sum_{i=1}^{v} \alpha^{i_v j} = \sum_{i=1}^{v} X_i^j \quad \text{Q.E.D.}$$

It is interesting to note that for $t = 1$, if the error occurs, for example, in the component numbered X_1, then the result of the parity check calculation is exactly $S_1 = X_1$ which is the Galois field binary code for the error position number. This is exactly analogous to the method of error-correction for Hamming codes in which the parity check calculation gives the ordinary binary code for the position of the error. In this sense the Bose-Chaudhuri codes for $t = 1$ are equivalent to the Hamming single-error correcting code.

The S_j are the power sum symmetric functions.[5] Thus the parity checks give the first t odd power sum symmetric functions. The first t even ones can be found from the fact that modulo 2, $(a + b)^2 = a^2 + b^2$, and hence

$$S_1^2 = \left[\sum_{i=1}^{v} X_i\right]^2 = \sum_{i=1}^{v} X_i^2 = S_2. \tag{4}$$

Similarly, $S_4 = S_1^4$, $S_6 = S_3^2$, etc.

Suppose that there are t errors. Then the error position numbers $X_1 \cdots X_t$ satisfy the equations

$$S_j = \sum_{i=1}^{t} X_i^j \qquad j = 1, 3, \cdots 2t - 1.$$

This is a set of t equations in t unknowns, the X_i. The solution would tell the positions of the errors. It appears impossible to solve the equations by any direct method, and trying all combinations of t of the $2^m - 1$ field elements would require too many computations. There is, however, an interesting compromise.

The elementary symmetric functions σ_i are related to the power sum symmetric functions S_i by Newton's identities:[5]

$$\left.\begin{aligned}
&S_1 - \sigma_1 = 0\\
&S_2 - S_1\sigma_1 + 2\sigma_2 = 0\\
&S_3 - S_2\sigma_1 + S_1\sigma_2 - 3\sigma_3 = 0\\
&S_4 - S_3\sigma_1 + S_2\sigma_2 - S_1\sigma_3 + 4\sigma_4 = 0\\
&S_5 - S_4\sigma_1 + S_3\sigma_2 - S_2\sigma_3 + S_1\sigma_4 - 5\sigma_5 = 0\\
&\cdots \quad \text{etc.}
\end{aligned}\right\} \tag{5}$$

If it is possible to solve Newton's identities for the elementary symmetric functions σ_i, the error position numbers must satisfy the equation

$$X^t - \sigma_1 X^{t-1} + \sigma_2 X^{t-2} \cdots \pm \sigma_t$$

$$= (X - X_1)(X - X_2) \cdots (X - X_t) = 0. \tag{6}$$

Eq. (6) can be solved effectively by merely substituting each of the $n = 2^m - 1$ field elements into the equation. For each digit in the received vector, the corresponding $GF(2^m)$ element is substituted in the equation. If the equation is satisfied, this bit is wrong and must be changed. If the equation is not satisfied, the bit is correct.

[5] See, for example, van der Waerden, footnote 8; J. Riordan, "An Introduction to Combinatorial Analysis," John Wiley and Sons, Inc., New York, N. Y., 1958; T. Muir and W. H. Metzler, "A Treatise on the Theory of Determinants," ch. 21, 1930; or any book on the Theory of Equations.

The proof that it is indeed possible to solve for the ordinary symmetric functions from the power sum symmetric functions is given by the following theorem:[6]

Theorem 1: The $k \times k$ matrix

$$M_k = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ S_2 & S_1 & 1 & 0 & \cdots & 0 \\ S_4 & S_3 & S_2 & S_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \\ S_{2k-4} & S_{2k-5} & S_{2k-6} & S_{2k-7} & & S_{k-3} \\ S_{2k-2} & S_{2k-3} & S_{2k-4} & S_{2k-5} & \cdots & S_{k-1} \end{bmatrix}$$

is nonsingular if power sum symmetric functions S_j are power sums of k or $k - 1$ distinct field elements, and is singular if the S_j are power sums of fewer than $k - 1$ distinct field elements.

The proof requires the following two lemmas:

Lemma 2: If the S_j are power sums of $v \leq k - 2$ distinct field elements, M_k is singular.

Proof:

$$M_k \begin{bmatrix} 0 \\ 1 \\ \sigma_1 \\ \cdot \\ \cdot \\ \cdot \\ \sigma_{k-2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

by Newton's identities, (5), and thus M_k has a nontrivial null space and must be singular. Q.E.D.

Lemma 3: If the S_j are power sums of k indeterminants $X_1, \cdots, X_k$, then the determinant

$$| M_k | = \prod_{i<i} (X_i + X_j).$$

Proof: If $X_i = X_j$, all of the power sums contain two identical terms, which cancel because the field has characteristic 2 (*i.e.*, $2 = 0$). Then it is just as if there were no more than $k - 2$ distinct elements used in forming the power sums, and, by Lemma 2, the determinant is zero. Therefore, $X_i + X_j$ is a factor of the determinant, for all i and j, and the left-hand side must be divisible by the right-hand side. It is easy to check that the left-hand side is homogeneous of degree $k(k - 1)/2$, the same as the right-hand side, and therefore they must differ at most by a constant factor.

To determine the constant factor, a single special case suffices. If k is odd, let the X_i be the roots of the equation

$$X^k - 1 = 0.$$

Then

$$\begin{aligned} \sum X_i^j = S_j &= 0 \quad \text{if} \quad j \equiv 0 \mod t, \\ &= 1 \quad \text{if} \quad j \equiv 0 \mod t. \end{aligned}$$

There will be exactly one 1 in each row and each column and it follows that $|M_k| = 1$ in this case. For k even, letting the X_i be all of the roots of the equation

$$X^k - X = 0$$

gives the same result. The constant factor, which could be only 0 or 1, must be 1.

Now Theorem 1 follows from the fact that if the determinant $|M_k|$ is zero it must be that some $X_i = X_j$. Since all of the nonzero X_i are distinct, $X_i = X_j = 0$, and there were fewer than $k - 1$ errors. Q.E.D.

If there are actually $t - 1$ errors, it can be seen from Newton's identities, Cramer's Rule and Theorem 1 that the solution for the σ's will yield $\sigma_t = 0$. The corresponding polynomial equation will have zero as one root.

Now let us review the error-correcting procedure. The t-error correcting Bose-Chaudhuri codes give, as the parity checks on received sequences, the odd power-sum symmetric functions up to S_{2t-1} and the intermediate even functions can be calculated simply from these. If it is assumed that no more than t errors occur, then by Theorem 1, with $k = t$, it is either possible to solve for the error position numbers, or there are $t - 2$ or fewer errors. In the latter case, $\sigma_{t-1} = \sigma_t = 0$, and two equations can be dropped, giving a set of $t - 2$ equations in $t - 2$ unknowns to which Theorem 1 can be applied again. Eventually, if there were any errors at all, a set of equations that can be solved for the elementary symmetric functions of the error-position numbers will be found.

The correction procedure consists of three phases:

1) calculate the parity checks and the even numbered S_j;

2) from these, calculate the elementary symmetric functions σ_i; and

3) finally, substitute each field element into the equation

$$X^t + \sigma_1 X^{t-1} + \sigma_2 X^{t-2} \cdots + \sigma_t = 0. \tag{7}$$

Those field elements which satisfy this equation correspond to error positions.

The second step involves a certain amount of trial and error because it is possible to solve the equations and obtain correct solutions only when the number of equations used equals or exceeds by one the number of errors that actually occur. This step might be carried out, as an alternative to the procedure described in the preceding paragraph, by starting with the assumption that two errors occurred, solving, and checking the solution. If the solution doesn't check, four errors would be assumed, and so forth. When a set of answers that checks occurs, it must be the correct solution.

[6] Similar results for a real field appear, for example, in H. O. Faulkes, "Theorems of Kakeya and Polya on Power sums," *Math. Z.*, vol. 65, pp. 345–352; 1956.

If it is assumed that the length n of the code approaches infinity and that the number of errors corrected t is a fixed fraction of n, the number of operations required for error correction can be crudely estimated as follows. The first phase, calculating parity checks, requires a number of operations proportional to the number of digits multiplied by the number of parity checks, or no more than nmt operations. This quantity nmt is proportional to n^2 log n. The second phase requires solving a $t \times t$ set of equations. The number of operations for this task is typically proportional to t^3, but it may have to be done $t/2$ times. This will increase in the limit no faster than n^4. Finally, substituting in a t-degree polynomial requires t multiplications and t additions of m digit numbers, and must be done n times, so that 2 tmn is a rough estimate of the number of operations. This again would vary as n^2 log n. Thus, the total number of operations certainly would increase as a small power of n.

Consider, as an example, the code corresponding to the matrix in (2), which corrects triple errors. The appropriate equations are

$$\begin{aligned} &S_1 + \sigma_1 = 0, \\ &S_3 + S_2\sigma_1 + S_1\sigma_2 + \sigma_3 = 0, \quad \text{and} \\ &S_5 + S_4\sigma_1 + S_3\sigma_2 + S_2\sigma_3 = 0. \end{aligned} \tag{8}$$

The parity checks for the received vectors give S_1, S_3, and S_5. $S_2 = S_1^2$, and $S_4 = S_1^4$. Solving for the σ's gives

$$\begin{aligned} &\sigma_1 = S_1, \qquad \sigma_2 = (S_1^2S_3 + S_5)/(S_1^3 + S_3) \quad \text{and} \\ &\sigma_3 = (S_1S_5 + S_3^2 + S_1^3S_3 + S_1^6)/(S_1^3 + S_3), \end{aligned} \tag{9}$$

provided that $S_1^3 + S_3 \neq 0$. If there is only one error $S_1^3 + S_3 = 0$. Furthermore, if $S_1^3 + S_3 = 0$, the Newton's identities yield $\sigma_3 = \sigma_1\sigma_2$, and the equation

$$\begin{aligned} &X^3 + \sigma_1X^2 + \sigma_2X + \sigma_3 \\ &\quad = X^3 + \sigma_1X^2 + \sigma_2X + \sigma_1\sigma_2 \\ &\quad = (X + \sigma_1)(X^2 + \sigma_2) = (X + \sigma_1)(X + \sqrt{\sigma_2})^2 = 0 \end{aligned}$$

has two equal roots, which must be zero, and therefore there is only one error.

As a numerical example, suppose that the vector of all zeros is transmitted, and that errors occur in the 2nd, 5th, and 7th positions. Then

$$r = (0\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)$$

$$r \times M = (1\ 0\ 1\ 1\quad 1\ 1\ 1\ 1\quad 1\ 0\ 0\ 0)$$

$$S_1 = (1\ 0\ 1\ 1) \qquad S_3 = (1\ 1\ 1\ 1) \qquad S_5 = (1\ 0\ 0\ 0).$$

Referring to Table I, one finds

$$S_2 = S_1^2 = (1\ 0\ 1\ 1)^2 = (\alpha^{13})^2 = \alpha^{26} = \alpha^{11} = (0\ 1\ 1\ 1)$$

and

$$S_4 = S_1^4 = \alpha^{52} = \alpha^7 = (1\ 1\ 0\ 1).$$

Then,

$$\begin{aligned} S_3 + S_1^3 &= (1\ 0\ 1\ 0) \neq 0, \\ \sigma_1 = S_1 &= (1\ 0\ 1\ 1) = \alpha^{13}, \\ \sigma_2 = (S_1^2S_3 + S_5)/S_3 + S_1^3 &= (0\ 0\ 1\ 0)/(1\ 0\ 1\ 0) \\ &= \alpha^2/\alpha^8 = \alpha^9/\alpha^{15} = \alpha^9. \end{aligned}$$

Similarly,

$$\sigma_3 = \alpha^{11}.$$

It is then easy to verify that the equation,

$$X^3 + \alpha^{13}X^2 + \alpha^9X + \alpha^{11} = 0,$$

is satisfied by the three values $X = \alpha$, α^4, and α^6, and only these. These correspond to the errors in r.

Some Properties of Cyclic Codes and Shift Register Generators

Codes for which the code points comprise a cyclic subspace of vectors of zeros and ones have been studied recently by Prange,[4] and, along with theoretical results, he found several efficient codes that can be decoded easily. He has noted that the codes can be coded with the use of a shift-register generator.[7] In this section, some of the theory of cyclic codes and linear recurrent sequences is reviewed briefly from a point of view that is especially well adapted to the study of the Bose-Chaudhuri codes.

A subset C of vectors of n binary digits is called a *cyclic subspace* if it has the following two properties:

1) If v_1 and v_2 are in C, their sum modulo 2 is also in C; that is, C is a subspace, or subgroup; and
2) If $v = (a_0, a_1, \cdots, a_{n-1})$ is in C, the vector $v^1 = (a_{n-1}, a_0, a_1, \cdots, a_{n-2})$ obtained by shifting v cyclically one place is also in C.

Let R_n denote the set of all polynomials

$$a_0 + a_1X \cdots + a_{n-1}X^{n-1}$$

of degree less than n with coefficients 1 and 0. They form a group under modulo 2 addition. Multiplication can be defined modulo $X^n - 1$; that is, these polynomials can be multiplied in the ordinary way, modulo 2, and then reduced again to polynomials of degree less than n by the use of the equation $X^n = 1$. Then R_n is a ring in the mathematical sense. A subset I of R_n is called an *ideal*[8] if it satisfies the following two properties:

1) I is a subgroup of R_n; and
2) if $p(X)$ is in I and $a(X)$ is R_n, then the product $p(X)$ $a(X)$ is in I.

[7] N. Zierler, "Linear recurring sequences," *J. Soc. Ind. Appl. Math.*, vol. 7, pp. 31–48; March, 1959.

[8] Galois fields and other aspects of algebra used in this paper are treated in many books on modern algebra. See, for example, A. A. Albert, "Fundamental Concepts of Modern Algebra," University of Chicago Press, Chicago, Ill., 1956; G. Birkoff and S. MacLane, "A Survey of Modern Algebra," The Macmillan Co., New York, N. Y., 1953; B. L. van der Waerden, "Modern Algebra," F. Ungar Publishing Co., New York, N. Y., vol. 1 and 2, 1949, 1950.

Considering polynomials $p(X) = a_0 + a_1X \cdots + a_{n-1}X^{n-1}$ to be vectors $(a_0, a_1, \cdots, a_{n-1})$, a cyclic shift is the same as multiplication by X modulo $X^n - 1$. Therefore, *every ideal is a cyclic subspace.* Conversely, if $p(X)$ is in a cyclic subspace C, so is $Xp(X)$. It follows that $X^ip(X)$ must also be in C, and since C is a subspace,

$$\sum_i c_iX^ip(X) = p(X) \sum_i c_iX^i$$

must also be in C. Thus, if $p(X)$ is in C, so is the product of $p(X)$ and any polynomial. Therefore, *every cyclic subspace is an ideal.*

The important but well-known properties of ideals given in the following three lemmas and two theorems are proved here to make the paper self-contained.

Lemma 4: If $p(X)$ and $q(X)$ are in an ideal I, the greatest common divisor (GCD), $d(X)$, of $p(X)$ and $q(X)$ is in I.

This follows directly from the fact that it is always possible to express the $d(X)$ in the form

$$d(X) = a(X)p(X) + b(X)q(X)$$

where $a(X)$ and $b(X)$ are polynomials.

Lemma 5: All polynomials in an ideal I are multiples of the unique polynomial of least degree in I. (That is, every ideal is a principal ideal.)

Proof: Let $p(X)$ be a polynomial of least degree in I. Then, if $q(X)$ is any other polynomial in I, the greatest common divisor of $p(X)$ and $q(X)$ is in I. If $p(X)$ does not divide $q(X)$, then the greatest common divisor of $p(X)$ and $q(X)$ would have lower degree than $p(X)$, which is a contradiction. Therefore, every polynomial in I is divisible by $p(X)$. If $p_1(X)$ and $p_2(X)$ both have minimum degree, each must be divisible by the other, and hence they are equal.

The ideal consisting of all multiples of $p(X)$ is denoted $[p(X)]$. The polynomial of least degree in an ideal is called its generator.

Lemma 6: The generator $p(X)$ of an ideal is a factor of $X^n - 1$.

Proof: The GCD $d(X)$ of $p(X)$ and $X^n - 1$ can be expressed in the form

$$\begin{aligned} d(X) &= a(X)p(X) + b(X)(X^n - 1) \\ &\equiv a(X)p(X) \quad \text{mod} \quad X^n - 1; \end{aligned}$$

hence, $d(X)$ is in the ideal. But $p(X)$ is divisible by $d(X)$, and since $d(X)$ is in the ideal, $d(X)$ is divisible by $p(X)$. Hence, $p(X) = d(X)$.

These results can be summarized as follows:

Theorem 2: A set of polynomials is an ideal in the ring of polynomials modulo $X^n - 1$ if and only if it consists of all multiples of degree less than n of a factor of $X^n - 1$.

Corollary: If $p(X)$ is a polynomial of degree k which divides into $X^n - 1$, $[p(X)]$ is a vector space of dimension $n - k$.

Proof: The elements of $[p(X)]$ are of the form $c(X)p(X)$ where $c(X)$ is an arbitrary polynomial of degree less than $n - k$. Then the $n - k$ coefficients of $c(X)$ are arbitrary.

Theorem 3: If $p(X)\ q(X) = X^n - 1$, the ideals $[p(X)]$ and $[q(X)]$ are null spaces of each other. That is, a polynomial $p_1(X)$ is in $[p(X)]$ if, and only if, $p_1(X)\ q_1(X) = 0$ modulo $(X^n - 1)$ for every polynomial $q_1(X)$ in $[q(X)]$.

Proof: Since $p_1(X)$ is in $[p(X)]$, $p_1(X)$ is a multiple of $p(X)$, for example, $a(X)\ p(X)$. Similarly, $q_1(X) = b(X)\ q(X)$. Then $p_1(X)\ q_1(X) = a(X)\ b(X)\ (X^n - 1) = 0$. Conversely, if $p_1(X)\ q(X) = 0$, then $p_1(X)\ q(X)$ must be a multiple of $X^n - 1$, and $p_1(X)$ must be a multiple of $(X^n - 1)/q(X) = p(X)$.

Note that the fact that the product of two polynomials is zero implies that the dot product of the corresponding two vectors is zero, if in one of them the order of the components is reversed. That is, if

$$(a_0 + a_1X \cdots + a_{n-1}X^{n-1})(b_0 + b_1X \cdots + b_{n-1}X^{n-1}) = 0$$

then

$$\begin{aligned} &(a_0, a_1 \cdots a_{n-1})\cdot(b_{n-1}, b_{n-2}, \cdots b_1, b_0) \\ &\qquad = a_0b_{n-1} + a_1b_{n-2} \cdots + a_{n-1}b_0 = 0, \end{aligned}$$

since this is the coefficient of X^{n-1} in the product of the polynomials. Hence, if $[p(x)]$ and $[q(x)]$ are null spaces of each other, the corresponding vector-spaces are null-spaces of each other provided that the order of components in the vectors of one of these is reversed.

Now let us consider a recursion relation (or difference equation) of the form

$$\sum_{j=0}^{k} a_jR_{i-j} = 0, \qquad (10a)$$

or

$$R_i = \sum_{j=1}^{k} a_jR_{i-j} \qquad a_0 = a_k = 1. \qquad (10b)$$

The solution of these equations for given coefficients a_n will be a sequence of binary digits, $\{R_i\}$. Given the digits $R_0, \cdots, R_{k-1}$, (10) is the rule for calculations R_k, then R_{k+1}, and so forth. Also, the sum of two solutions is again a solution because the equation is linear. Therefore, the solutions form a vector space of dimension k. The solutions are characterized in the following theorem.

Theorem 4: Let $p(X) = \sum_{j=0}^{k} a_jX^j$, $a_0 = a_k = 1$, and let n be the smallest integer for which $X^n - 1$ is divisible by $p(X)$. Let $q(X) = (X^n - 1)/p(X)$. Then the solutions of the difference equation

$$R_i = \sum_{j=1}^{k} a_jR_{i-j}$$

are periodic of period n, and the set made up of the first period of each possible solution, considered as polynomials, is the ideal $[q(X)]$.

Proof: That any vector taken from $[q(X)]$ is a solution can be seen by multiplying a polynomial from $[q(X)]$, for example, $q_1(X)$, by $p(X)$. The digits in the product are formed by the summation in (10a), and, since the product is zero, (10a) is satisfied. Therefore, any sequence formed by repetition of a vector taken from $[q(X)]$ is a solution of (10). Since $q(X) = X^n - 1/p(X)$ has degree $n - k$, then $[q(X)]$ has dimension k, by the corollary to Theorem 2. This is the same as the dimension of the space of solutions, and therefore $[q(X)]$ must include all solutions.

The Cyclic Structure of the Bose-Chaudhuri Codes

It is shown in this section that the Bose-Chaudhuri codes are examples of cyclic codes as studied by Prange.[4] As such they can be generated with very simple equipment, as is illustrated for the (15,5) code in the next section. Out of this theory also comes a better estimate of the number of parity check digits required to correct a given number of errors.

By the alternative definition of the Bose-Chaudhuri codes given in the second section of this paper, a code consists of all polynomials $f(X)$ which have $\alpha, \alpha^3, \cdots, \alpha^{2t-1}$ as roots. Each element α^i of the field is a root of a unique irreducible polynomial $p_i(X)$ of minimum degree. Then $f(X)$ must be divisible by each of the polynomials $p_1(X), p_3(X), \cdots, p_{2t-1}(X)$ and, hence, by their least common multiple:[9]

$$f(X) = \operatorname*{LCM}_{i=1,3,\cdots,2t-1} [p_i(X)]. \tag{11}$$

Since each of the factors $p_i(X)$ is irreducible, the least common multiple of the $p_i(X)$ is simply the product of the polynomials $p_i(X)$, with the duplicates omitted. Duplications are quite possible; they will occur, in fact, for any α^i and α^j that are roots of the same polynomial $p_i(X)$. In other words, should α^i and α^j happen to be roots of the same irreducible polynomial, the columns in the parity check matrix will be dependent, although not necessarily identical. The parity checks produced by the column of powers of α^j will be satisfied if and only if the parity checks produced by the column of powers of α^i are satisfied, and thus one set or the other is unnecessary.

Finally, the set of all sequences that comprise the code can, by Theorem 4, be generated by a recursion relation defined by the polynomial $X^n - 1/f(X)$, and hence by a shift register generator.

At this point it is interesting to study the limiting cases of the minimum and maximum numbers of parity checks. It has already been noted that the nontrivial minimum is the Hamming code. On the other extreme, the last two columns which might be included in the parity check matrix are powers of $\alpha^{2m-2} = \alpha^{-1}$ and $\alpha^{2m-1} = 1$. The last one is a root of the irreducible polynomial $1 + x$ and the resulting code would be the ideal generated by $(1 + x^n)/(1 + x)$. This ideal consists of the zero vector and the vector of all ones, so the code is the trivial repetition of a single information digit $n = 2^m - 1$ times. If α is a primitive element, so is α^{-1}, and therefore the irreducible polynomial of which α^{-1} is a root is primitive. It can be shown then that when only the last two columns, corresponding to α^{-1} and 1, are omitted from the parity check matrix, the resulting code consists of a maximal length sequence, all its shifts, all complements, and a sequence of all 1's, which is then the code studied by San Soucie and Green.[10] This code can also be shown to be equivalent to the Reed-Muller first-order code with any one digit dropped.[11]

It is possible to predict easily which powers of α are roots of the same polynomial, and thus, incidentally, find the degree of the polynomial of which α^i is a root. The method is based on the fact that if a is a root of $f(X)$, then a^2 is also, since $f(a^2) = [f(a)]^2 = 0$. It turns out that $a, a^2, a^4, a^8, \cdots$ are, in fact, all of the roots. In Table II information is given for $m = 4$ and 5. Note that in the first case, $\alpha^{15} = 1$; and in the second, $\alpha^{31} = 1$.

The code for $m = 4$, $t = 3$ has for its generator, by (11),

$$f(X) = p(X)p_3(X)p_5(X)$$

and therefore has $4 + 4 + 2 = 10$ parity checks, and 5 information places. The code for $m = 5$, $t = 5$ has

$$f(X) = p(X)p_3(X)p_5(X)p_7(X)$$

for its generator, and therefore has 20 parity checks. All codes for $m = 4$ and 5 are listed in Table III.

TABLE II
Roots of Polynomials $p_j(X)$

Polynomial		Roots
$m = 4$	$p(X)$	$\alpha, \alpha^2, \alpha^4, \alpha^8$
	$p_3(X)$	$\alpha^3, \alpha^6, \alpha^{12}, \alpha^9$
	$p_5(X)$	α^5, α^{10} $(\alpha^{20} = \alpha^5)$
	$p_7(X)$	$\alpha^7, \alpha^{14}, \alpha^{13}, \alpha^{11}$
$m = 5$	$p(X)$	$\alpha, \alpha^2, \alpha^4, \alpha^8, \alpha^{16}$
	$p_3(X)$	$\alpha^3, \alpha^6, \alpha^{12}, \alpha^{24}, \alpha^{17}$
	$p_5(X)$	$\alpha^5, \alpha^{10}, \alpha^{20}, \alpha^9, \alpha^{18}$
	$p_7(X)$	$\alpha^7, \alpha^{14}, \alpha^{28}, \alpha^{25}, \alpha^{19}$
	$p_9(X) = p_5(X)$	
	$p_{11}(X)$	$\alpha^{11}, \alpha^{22}, \alpha^{13}, \alpha^{26}, \alpha^{21}$
	$p_{13}(X) = p_{11}(X)$	
	$p_{15}(X)$	$\alpha^{15}, \alpha^{30}, \alpha^{29}, \alpha^{27}, \alpha^{23}$

TABLE III
Rate and Error Correction Ability of Bose-Chaudhuri Codes for $m = 4$ and 5

Length of Code Words	Number of Parity Checks	Number of Information Places	Number of Errors Corrected
n	$n - k$	k	t
15	4	11	1
15	8	7	2
15	10	5	3
31	5	26	1
31	10	21	2
31	15	16	3
31	20	11	5
31	25	6	7

[9] See, for example, Birkhoff and MacLane, *op. cit.*, p. 396.

[10] J. H. Green, Jr. and R. L. San Soucie, "An error-correcting encoder and decoder of high efficiency," Proc. IRE, vol. 46, pp. 1741–1744; October, 1958.

[11] N. Zierler, "On a variation of the first-order Reed-Muller Codes," Lincoln Laboratory Group Rept. 34-80; October, 1958.

Code parameters for some larger codes were calculated on the IBM 704 computer. The results are plotted in Fig. 1. The vertical axis represents rate (percentage of all digits available for information), and the horizontal axis represents the number of errors correctable as a percentage of the total number of digits. The dashed curve represents asymptotic values of a lower bound on the rate of the best code that corrects errors in a given percentage of the digits.[12] The curves drawn for the Bose-Chaudhuri codes for large n fall below the bound for the best code. In fact, it is shown in the Appendix that they approach zero as the length of the code increases indefinitely. This may mean that these codes are truly not optimum, or it may mean that the number of errors correctable by the procedure given in this paper is not the total number of errors correctable by Bose-Chaudhuri codes in the case of very long codes.[13]

The polynomial $p(X)$ can be any primitive polynomial of degree m. The other polynomials $p_i(X)$ are determined by the particular choice of $p(X)$, and the question arises as to how they may be calculated. One simple method is based on the fact that every element of $GF(2^m)$ is a root of the polynomial $X^{2m-1} - 1$. Therefore, each element is a root of one of the factors of $X^{2m-1} - 1$. One needs only to factor this polynomial and test to see which factor has X^i as a root. The following alternative method is useful. It has been noted that the degree m_i of $p_i(X)$ can be easily determined. Then if

$$p_i(X) = a_0 + a_1X + \cdots + a_{m-1}X^{m_i-1} + X^{m_i},$$

since α^i is a root of $p_i(X)$,

$$0 = a_0\alpha^0 + a_1\alpha^1 + \cdots + a_{m-1}\alpha^{m_i-1} + \alpha^{m_i},$$

and if α^i is written as a vector with m components, the resulting set of linear equations can be solved for the coefficients a_i of $p_i(X)$.

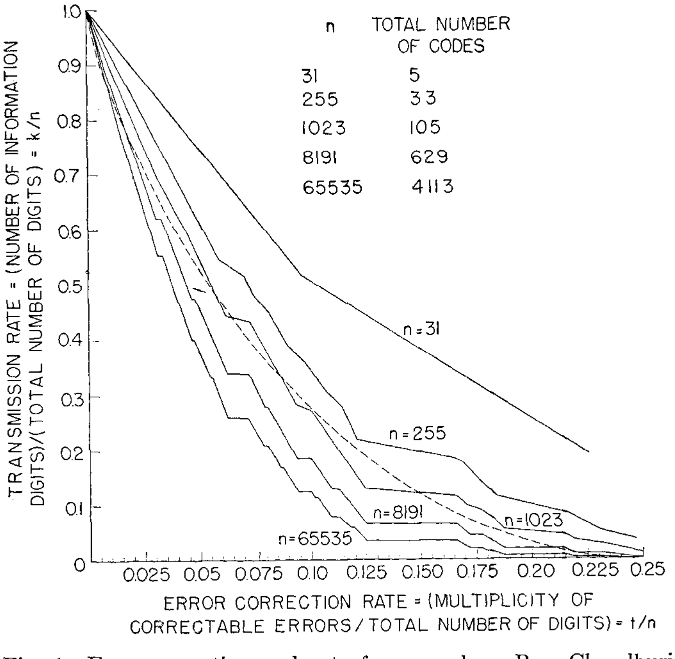


Fig. 1—Error correction and rate for some long Bose-Chaudhuri codes. (Dashed curve is asymptotic lower bound for the rate for the best binary code as given by Gilbert.)

There is also an explicit formula

$$p(X^{1/j})p(\alpha X^{1/j}) \cdots p(\alpha^{j-1}X^{1/j})$$

where α is a primitive jth root of unity. It can be shown that when the multiplication is carried out only integral powers of X remain, and these have only ones or zeros as coefficients.

Consider again the sample code discussed by Bose and Chaudhuri. The irreducible factors of $X^{15} - 1$ are

$$X^{15} - 1 = (X - 1)(X^2 + X + 1)(X^4 + X^3 + X^2 + X + 1)(X^4 + X^3 + 1)(X^4 + X + 1).$$

A root of the last factor was taken as α; and thus

$$p(X) = X^4 + X + 1.$$

Then α^3 satisfies the equation $X^5 - 1 = 0$, since $\alpha^{15} = 1$. But $X^5 - 1 = (X - 1)\,(X^4 + X^3 + X^2 + X + 1)$, and since α^3 is not a root of the first factor, it must be a root of the second. Similarly, α^5 satisfies $X^3 - 1 = 0 = (X - 1)\,(X^2 + X + 1)$, and so α^5 is a root of $X^2 + X + 1$. The fact that this has degree 2 ties in with the observation that the column of powers of α^5 contained only two independent parity checks.

All code points must be multiples, then, of

$$\begin{aligned} f(X) &= p(X)p_3(X)p_5(X) \\ &= (1 + X + X^4)(1 + X + X^2 + X^3 + X^4) \\ &\qquad \cdot(1 + X + X^2) \\ &= 1 + X + X^2 + X^4 + X^5 + X^8 + X^{10} \\ &= (1\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0), \end{aligned} \tag{12}$$

and it can easily be checked that this vector, any cyclic permutation of it, and any sum of permutations, actually do satisfy the parity checks defined by the matrix M in (2).

[12] E. N. Gilbert, "A comparison of signaling alphabets," *Bell Sys. Tech. J.*, vol. 31, pp. 504–522; May, 1952.

[13] I have found with the aid of the IBM 704 that the Bose-Chaudhuri two-error correcting codes for $m = 4$ and 5 correct some triple errors and nothing beyond and are therefore optimum. The three-error correcting code for $m = 4$ corrects 420 quadruple and 28 quintuple error patterns and is optimum. The three-error correcting code for $m = S$ corrects 13,020 quadruples and 14,756 quintuples and nothing beyond—this seems good but has not been proved optimum. (See A. B. Fontaine and W. W. Peterson, "Group code equivalence and optimum codes," IRE Trans. on Information Theory, vol. IT-5, pp. 60–70; May, 1959.) Thus, any non-optimum behavior of these codes occurs only in codes so large that they are difficult to analyze by looking at code words themselves or searching for coset leaders even with the aid of a computer.

Mechanizing the Coding and Error-Correction

Shift registers with feedback corrections can be used in a number of ways in mechanizing coding and error-correction procedures. The following uses will be discussed in this section:

1) coding using a shift register with one stage for each information digit in the code,
2) coding using a shift register with one stage for each parity check digit in the code,
3) counting in the Galois field code,
4) multiplying and dividing Galois field elements, and
5) calculating parity checks on received vectors.

Both the methods of coding apply to any cyclic code. The methods will be illustrated using the Bose-Chaudhuri (15, 5) code described by the matrix M in (2).

Every cyclic code is an ideal generated by some polynomial $f(X)$, *i.e.*, a polynomial is a code vector if and only if it is divisible by $f(X)$. This means that, by Theorem 4, a vector is a code vector if and only if it satisfies the recursion relation corresponding to the polynomial $(X^n - 1)/f(X)$. For the code used as an example, by (12),

$$f(X) = 1 + X + X^2 + X^4 + X^5 + X^8 + X^{10}$$
$$(1 - X^{15})/f(X) = 1 + X + X^3 + X^5.$$

Then every sequence satisfying the recursion relation

$$R_j = R_{j-1} + R_{j-3} + R_{j-5}$$

is a code point, and conversely. Such sequences can be generated by putting information digits in the shift register generator shown in Fig. 2 and shifting 15 times. The first five digits coming out will be information digits, and the next ten digits will be a set of parity checks which make the whole sequence a code point. The symbols come out of this encoder low order digits first. The order can be reversed by reversing the order of the shift register feedback connections.

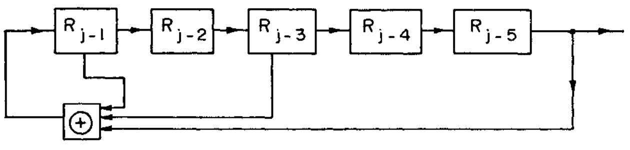


Fig. 2—A shift register for encoding the Bose-Chaudhuri (15,5) code.

A second method of coding is based again on the fact that the coded vector must be, considered as a polynomial, a multiple of $f(X)$. Let $t_0(X)$ be a polynomial in which the k coefficients of the terms involving X^{n-1} through X^{n-k} are arbitrary information digits, and the coefficients of lower order terms are zero. This corresponds to a vector in which the first $n - k$ components are zero, the last k digits arbitrary information digits. Then $t_0(X)$ can be divided by $f(X)$ to produce a quotient and a remainder

$$t_0(X) = f(X)q(X) + r(X),$$

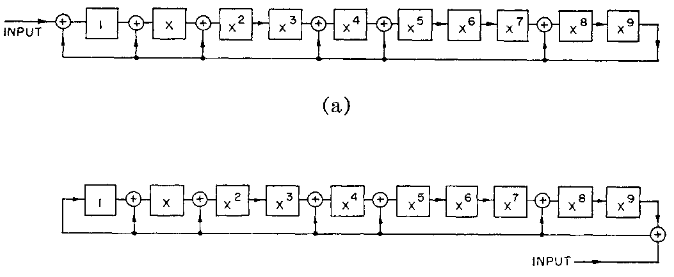
where $r(X)$ has degree less than $(n - k)$, which is the degree of $f(X)$. Then

$$t_0(X) + r(X) = f(X)q(X)$$

and, hence, $t_0(X) + r(X)$ is a code point. But $r(X)$ corresponds to a vector in which all components except the first $n - k$ are zero, since $r(X)$ has degree less than $n - k$. Thus, the sum consists of $n - k$ check digits, the coefficients of $r(X)$, and k information digits, the coefficients of $t_0(X)$.

The next problem is to calculate $r(X)$. In general, the calculation of the remainder after division by a polynomial can be accomplished with a shift register. The method is illustrated in Fig. 3(a). Assuming the divisor is the $f(X)$ for the code used in the example, *i.e.*, $1 + X + X^2 + X^4 + X^5 + X^8 + X^{10}$, the operation of the circuit can be understood as follows: The answer is the same as results from reducing the dividend modulo $f(X)$. This means that the dividend polynomial should be reduced to a polynomial of degree less than 10 using the relation

$$X^{10} = 1 + X + X^2 + X^4 + X^5 + X^8. \qquad (13)$$



(a)

(b)

Fig. 3—Shift register for calculating residues modulo $f(X) = 1 + X + X^2 + X^4 + X^5 + X^8 + X^{10}$. (a) Basic circuit; (b) basic circuit with automatic premultiplication by X^{10}.

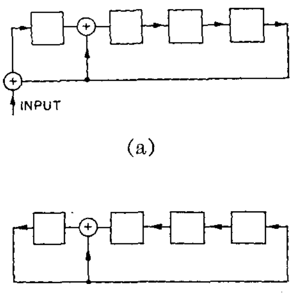
Now assume that a single one is shifted into the low-order position and then shifted right a number of times. Thinking of the contents of the register as a polynomial with low order digits at the left, each shift corresponds to multiplying by X, at least until a shift out of the high-order position. A one in the high-order position corresponds to X^9, and shifting it out makes it X^{10}. This results in the circuit in adding into the lower order positions the equivalent of X^{10} given in (13), and, hence, in this case the shift still corresponds to multiplying by X and modulo $f(X)$. Thus, successive shifts give successive powers of X modulo $f(X)$.

Now this is a linear device, and a polynomial (which is the sum of powers of X) can be reduced modulo $f(X)$ by shifting it into the device, high power terms first, until the constant term is shifted into the low-order position.

In using this device for calculating the $r(X)$ in (17), the modification shown in Fig. 3(b) can be made to avoid the last $n - k$ shifts which would add $n - k$ zeros into the low-order positions. It amounts to multiplying the input digits by $X^{n-k} = X^{10}$ before adding.

The procedure for coding is then to shift all the information digits into the device in Fig. 3(a) or 3(b). If the device in Fig. 3(a) is used, $n - k$ more shifts must be made with no input. Then the correct check digits remain in the register and should simply follow the information digits, high order digits first, to make a complete code vector. Note that the number of stages in this shift register is $n - k$, while the shift register shown in Fig. 2 has k stages.

A counter which counts in terms of Galois field elements is shown in Fig. 4(a). It works on the same principle as the device shown in Fig. 3(a), but using the primitive polynomial $p(X) = X^4 + X + 1$ of which α is a root. If a 1 is placed in the low-order position, successive shifts give successive powers of α using the relation $\alpha^4 = \alpha + 1$, and these are exactly the representations of $GF(2^4)$ elements given in Table I.



(a)

(b)

Fig. 4—Galois field counters for $GF(2^4)$. (a) Increasing powers of α; and (b) decreasing powers of α.

In the device shown in Fig. 3(b), a left shift corresponds to division by α and a 1 shifted out of the low order end α^{-1} is replaced by its equivalent $1 + \alpha^3$. Thus, this device can count down, or give Galois field elements in reverse order. A multiplier can be mechanized by putting one factor in a device A like that shown in Fig. 3(a), the other in a device B like that shown in Fig. 3(b). Then both devices are shifted until the code for 1 appears in device B. The product then appears in A. Division can be done in an anologous manner. Multiplication can also be done in a manner analogous to that used in digital computers with a shift register such as that shown in Fig. 3(a) used in place of an accumulator.

The parity checks corresponding to the first column of Galois field elements in the matrix M of (2) correspond to the Galois field representation of

$$r(\alpha) = r_0 + r_1\alpha + r_2\alpha^2 \cdots r_{2^m-2}\alpha^{2^m-2}.$$

This can be calculated by using the relation $\alpha^4 + \alpha + 1 = 0$ to eliminate terms of degree higher than 3 in α. This, in turn, is exactly what will result if the vector $(r_0, r_1, \cdots, r_2m_2)$ is shifted into the shift register shown in Fig. 3(a) high-order digits first. Note that shifting fifteen times multiplies by α^{15}, but $\alpha^{15} = 1$. Similarly, the device in Fig. 3(b) could be used with the low-order digits entering first.

Calculation of the other parity checks is slightly more complicated. It requires calculating $r(\alpha^j)$ for the first t odd values of j. The first step is to devise a shift register which automatically multiplies by α^j. The example $j = 5$ should make the principles clear. Note that

$$\begin{aligned} 1\cdot\alpha^5 &= \alpha^5 = \alpha + \alpha^2 \\ \alpha\cdot\alpha^5 &= \alpha^6 = \alpha^2 + \alpha^3 \\ \alpha^2\cdot\alpha^5 &= \alpha^7 = 1 + \alpha + \alpha^3 \\ \alpha^3\cdot\alpha^5 &= \alpha^8 = 1 + \alpha^2, \end{aligned}$$

so that

$$\begin{aligned} \alpha^5(a_0 + a_1\alpha + a_2\alpha^2 + a_3\alpha^3) \\ &= a_0(\alpha + \alpha^2) + a_1(\alpha^2 + \alpha^3) \\ &\quad + a_2(1 + \alpha + \alpha^3) + a_3(1 + \alpha^2) \\ &= (a_2 + a_3) + (a_0 + a_2)\alpha \\ &\quad + (a_0 + a_1 + a_3)\alpha^2 + (a_1 + a_2)\alpha^3. \end{aligned}$$

Thus, the new value of a_0 is the old $a_2 + a_3$, the new a_1 is the old $a_0 + a_2$, etc. A shift register with feedback connections shown in Fig. 5 will give this result. Then, if the received vector $(r_0, r_1, \cdots, r_2m_2)$ is shifted into this device, after fifteen shifts the result $r(\alpha^5)$ will remain in the register.

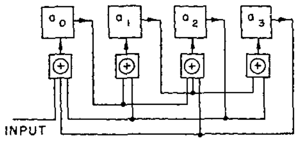


Fig. 5—A circuit for calculating the parity checks $r(\alpha^5)$.

Conclusion

Relatively simple coding and error-correcting methods have been described for the Bose-Chaudhuri codes. The study of coding and error-correction methods for these codes gives additional insight into the remarkable structure of the codes.

Appendix

A bound on the rate of Bose-Chaudhuri codes which correct $t = 2^\lambda$ errors is derived in this Appendix, and it is shown on the basis of this bound that if t is made a fixed fraction of n, the number of digits in the code, the rate must approach zero as n increases indefinitely.

This problem is purely number-theoretic, and can be formulated as follows: The quantity to be studied is the rate, which is the quotient of the number k of information digits and $n = 2^m - 1$, the total number of digits. Since there is one independent parity check for each distinct residue of $j2^i$ for $1 \leq j \leq 2t$, $0 \leq i < m$, the number of such residues in $n - k$. Since $2^m = 2^0$ modulo $2^m - 1$, the condition $0 \leq i < m$ can be replaced by $1 \leq i \leq m$. For convenience in what follows, j will be allowed to take on the value zero also; this adds one distinct residue.

Let $N(s)$ be the number of distinct residues of $j2^i$ for $0 \leq j < 2t = 2^{\lambda+1}$ and $m - s < i \leq m$. Then

$$n - k = N(m) - 1 \geq N(s) - 1 \quad \text{if} \quad s \leq m$$

and

$$k = n - N(m) + 1$$

$$= 2^m - N(m) \leq 2^m - N(s) \quad \text{if} \quad s \leq m. \qquad (14)$$

An equation for $N(s)$, valid only for $s \leq \lambda$, will be derived but this will give an upper bound on k by (14).

Consider first the residues for a particular value of i, $m - \lambda \leq i \leq m$. They can be arranged as follows:

$$\begin{array}{cccc}
0\cdot 2^i, & 1\cdot 2^i, & 2\cdot 2^i, \quad \cdots, & (2^{m-i} - 1)2^i \\
(2^{m-i} + 0)2^i, & (2^{m-i} + 1)2^i, & (2^{m-i} + 2)2^i, \quad \cdots, & (2\cdot 2^{m-i} - 1)2^i \\
(2\cdot 2^{m-i} + 0)2^i, & (2\cdot 2^{m-i} + 1)2^i, & (2\cdot 2^{m-i} + 2)2^i, \quad \cdots, & (3\cdot 2^{m-i} - 1)2^i \\
\vdots & \vdots & \vdots & \vdots \\
(2^{\lambda+1} - 2^{m-i} + 0)2^i, & (2^{\lambda+1} - 2^{m-i} + 1)2^i, & (2^{\lambda+1} - 2^{m-i} + 2)2^i, \quad \cdots, & (2^{\lambda+1} - 1)2^i.
\end{array}$$

In this array there are $2^{\lambda+1-m+i}$ rows. Since $2^m \equiv 1$, the array can be rewritten

$$\begin{array}{ccccc}
0, & 1\cdot 2^i, & 2\cdot 2^i, & \cdots, & (2^{m-i} - 1)2^i \\
1, & 1 + 1\cdot 2^i, & 1 + 2\cdot 2^i, & \cdots, & 1 + (2^{m-i} - 1)2^i \\
2, & 2 + 1\cdot 2^i, & 2 + 2\cdot 2^i, & \cdots, & 2 + (2^{m-i} - 1)2^i \\
\vdots & \vdots & \vdots & & \vdots \\
2^{\lambda+1+i-m} - 1, & (2^{\lambda+1+i-m} - 1) + 1\cdot 2^i, & (2^{\lambda+1+i-m} - 1) + 2\cdot 2^i, & \cdots, & 2^{\lambda+1+i-m} - 1 + (2^{m-i} - 1)2^i.
\end{array}$$

This consists exactly of 2^{m-i} sets of $2^{\lambda+1+i-m}$ successive numbers starting at each multiple of 2^i. The arrangement is shown graphically in Fig. 6.

The important facts can be seen clearly in Fig. 6 but are tedious to prove formally. For each i there are $2^{\lambda+1}$ residues and therefore, in particular, $N(1) = 2^{\lambda+1}$. Two adjacent columns in Fig. 6 have half their residues in common. In particular, $N(2) = 2^{\lambda+1} + 2^\lambda$. Now in adding the contributions to $N(s)$ for larger values of s it is necessary to determine exactly how many residues have occurred in all previous columns combined. There is one other case which must be considered besides the previous adjacent column. Note that the residues and nonresidues of $j\cdot 2^i$ for a particular value of i fall in blocks of $2^{\lambda+1+i-m}$ successive numbers. In determining which residues for a particular value of i, for example, i_0, have occurred before, each block of 2^{i_0+1} successive numbers is treated the same. Each will have two blocks of $2^{\lambda+1+i_0-m}$ residues. The first will already have been counted in the $i_0 + 1^{st}$ column. The fraction of the others to be omitted is the same as the fraction of blocks of length 2^{i_0+1} which were counted as residues for $i \geq i_0 + m - \lambda$, which is the same as $N(\lambda - i_0)/2^m$. Then, since $s = m - i_0$,

$$N(s) = N(s-1) + 2^\lambda\cdot[1 - 2^{-m}N(\lambda - m + s)] \qquad (15)$$

for $0 < s \leq \lambda$. [$N(s)$ should be considered zero for $s \leq 0$.]

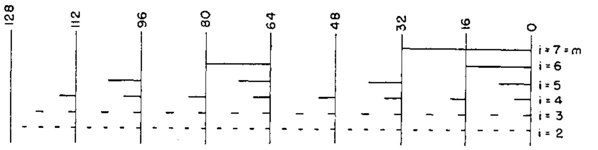


Fig. 6—Distribution of residues of $j2^i$($m = 7$, $\lambda = 4$).

Now let

$$R(s) = 1 - N(s)\cdot 2^{-m}.$$

Since $N(s)$ includes the zero residue, the actual number of parity digits is at least $N(s) - 1$. The actual number of information digits is at most $2^m - 1 - N(s) + 1 = 2^m - N(s)$. The actual rate would be at most $[2^m - N(s)]/(2^m - 1)$, but for large m, this is approximately $R(s)$. Then

$$N(s) = 2^m[1 - R(s)],$$

and substitution in (15) results in a difference equation for $R(s)$:

$$R(s) = R(s-1) - 2^{\lambda-m}R(s - m + \lambda) \qquad (16)$$

for $0 < s$. [$R(s)$ should be considered to be 1 for $s \leq 0$.] Clearly,

$$1 \geq R(s) \geq 0 \quad \text{for all} \quad s. \qquad (17)$$

It follows at once from (16) and (17) that $R(s)$ is nonincreasing. Now if there exists $\epsilon > 0$ such that $R(s) > \epsilon$ for all s, choose any $s_0 > m - \lambda + (2^{m-\lambda}/\epsilon)$. Then $R(s_0) = [R(s_0) - R(s_0 - 1)] + [R(s_0 - 1) - R(s_0 - 2)] + \cdots + [R(m - \lambda + 1) - R(m - \lambda)] + R(m - \lambda)$ trivially $= R(m - \lambda) - 2^{\lambda-m}[R(s_0 - m + \lambda) + R(s_0 - m + \lambda - \ ) + \cdots + R(1)]$ by (16) $< R(m - \lambda) - 2^{\lambda-m}(s_0 - m + \lambda)\epsilon$ by hyoothesis $< R(m - \lambda) - 1$ by choice of $s_0 < 0$ by half of (17), contradicting the other half, and proving that $R(S) \rightarrow 0$ as $s \rightarrow \infty$ must hold.

Now suppose that it is required that errors be corrected in a fraction 2^{-v} of the number of digits in a code word. Then

$$2^{-v} = 2^{\lambda}/2^m - 1 \approx 2^{\lambda-m},$$

so $v \approx m - \lambda$. Then, taking $s = \lambda$, $R(\lambda) = R(m - v)$ is an upper bound on the rate for a code with $2^m - 1$ digits. As m increases this approaches zero. Since rate is a monotone nonincreasing function of the number of errors correctable and the rate approaches zero for arbitrarily small fractions $t/n = 2^{-v}$, it must approach zero for any fraction $t/n > 0$.

Acknowledgment

I have benefited greatly from discussions with many people at the IBM Research Laboratory and at the Research Laboratory of Electronics at Massachusetts Institute of Technology. E. Prange, of the Air Force Cambridge Research Center, J. Griesmer and J. Selfridge of IBM, and M. P. Schutzenberger and S. Golomb at the Research Laboratory of Electronics were especially helpful.

Most of all, I am indebted to R. C. Bose of the University of North Carolina, for lecturing on his and Chaudhuri's fine work so soon after it was done and for the very stimulating discussion we had during his visit to the IBM Research Laboratory in August 1959.

Part of the computation work was done at the M.I.T. Computation Center.

Synchronization of Binary Messages*

E. N. GILBERT†

Summary—When messages are transmitted as blocks of binary digits, means of locating the beginnings of blocks are provided to keep the receiver in synchronism with the transmitter. Ordinarily, one uses a special synchronizing symbol (which is really a third kind of digit, neither 0 nor 1) for this purpose. The Morse code letter space and the teletype start and stop pulses are examples. If a special synchronizing digit is not available, its function may be served by a short sequence of binary digits P which is placed as a prefix to each block. The other digits must then be constrained to keep the sequence P from appearing within a block. If blocks of N digits (including the prefix P) are used, the prefix should be chosen to make large the number $G(N)$ of different blocks which satisfy the constraints. Lengthening the prefix decreases the number of "message digits" which remain in the block but also relaxes the constraints. Thus, for each N, there corresponds some optimum length of prefix.

For each prefix P, a generating function, a recurrence formula, and an asymptotic formula for large N are found for $G(N)$. Tables of $G(N)$ are given for all prefixes of four digits or fewer. Among all prefixes P of a given length A, the one for which $G(N)$ has the most rapid growth is $P = 11 \cdots 1$. However, for this choice of P, the table of values of $G(N)$ starts with small values; $11 \cdots 1$ does not become the best A-digit prefix until N is very large. At these values of N, the $(A + 1)$ − digit prefix $11 \cdots 10$ is still better. The tables suggest that, for any N, a best prefix can always be found in the form $11 \cdots 10$, for suitable A. Taking $P = 11 \cdots 10$ and $A = [\log_2 (N \log_2 e)]$ it is shown that $G(N)$ is roughly $0.35N^{-1}2^N$. This result is near optimal since no choice of P can make $G(N)$ exceed $N^{-1}2^N$.

I. Introduction

When block coding is used, some care must be taken to ensure that the transmitter and receiver stay in synchronism. For example, the Morse code letter spaces and the teletype stop and start pulses

* Received by the PGIT, December 20, 1959.

† Mathematical Res. Dept., Bell Telephone Labs., Murray Hill, N. J.