NRL Report 5993

An Application of Bose-Chaudhuri Error-Correcting Codes

F. A. Polkinghorn, Jr.
Radio Division

September 25, 1963

APPROVED FOR PUBLIC
RELEASE - DISTRIBUTION
UNLIMITED

U.S. NAVAL RESEARCH LABORATORY
Washington, D.C.

CONTENTS

Abstract 1
Problem Status 1
Authorization 1

INTRODUCTION 1

TECHNICAL CONSIDERATIONS 1

SUMMARY 3
REFERENCES 3

Appendix A – Implementation of Bose-Chaudhuri Codes on a Digital Computer 3

An Application of Bose-Chaudhuri Error Correcting Codes

FRANK A. POLKINGHORN, JR.
Radio Division

This report proposes a solution to the problem of the use of error correcting codes on a particular multichannel hf radio circuit with a slowly time varying probability of bit error and states in detail a computer algorithm for this solution. The conclusion is reached that the single, double, and triple error correcting Bose-Chaudhuri codes of block length 31, which have been stunted to a block length of 30, should be used. A small general purpose computer will be used to decode the blocks, possibly to compose a request for retransmission of uncorrectable errors, and to decide, on the basis of current error rates on a particular channel, the type of code to be used on that particular channel.

INTRODUCTION

A problem has arisen which involves the encoding and decoding of many channels, where each channel has an assumed independent, binary symmetric, time dependent probability of bit error. It is the purpose of this report to present a class of possible solutions, the Bose-Chaudhuri codes of block length 31, for this purpose. A general purpose computer-decoder will be used, in which case the time dependent situation can be handled by a fixed code header on each message to instruct the decoder that a particular code will follow and how this code will be used.

TECHNICAL CONSIDERATIONS

The first technical consideration is concerned with the criteria which a code must meet. Since the code in which we are interested will be used on a teletype system with a noncomputer backup mode, the code must consist of 5N information bits followed by 5M parity bits in a block. Next, the code must be relatively efficient, where the total message efficiency is the measurable quantity. Therefore, a block code whose length is long with respect to the variable message length is undesirable, as one expects to waste about one-half of the information bits in the last block. There is also a problem of decoding long blocks in real time, since the decode time is approximately linear with the number of bits in the block. For short codes the overhead time becomes excessive, and the block efficiency is low; hence, they too are undesirable. The third desirable feature would be the ability to correct the largest portion of the bit errors and still make the probability of an undetected error extremely small. Unfortunately, the statistics are not presently available, and one is limited to an educated guess as to the distribution of bit errors. Last, since it is possible that a general purpose encoder may not be available at the transmitter, the cost of special purpose encoders should be relatively small.

The current length of messages which will be used in the system narrows the choice of Bose-Chaudhuri codes to those of block length 31, 63, and 127 bits as possible block lengths. The power of the Bose-Chaudhuri codes on hf radio circuits has been demonstrated by A. B. Fontaine (1); he used a (255, 239) double error correcting code solely for error detection. On one 800-mile teletype circuit operating at 28.4 bits per second (bps), he estimated the undetected error rate to be one error in two years. On an 8000-mile teletype circuit operating at 85.9 bps and equipped with RCA's automatic request for repetition (ARQ) system, the estimated undetected error rate was one error in 0.31 year. It was expected that we could achieve similar results with a shorter block length and comparable detection capability plus some error correcting capability and a computer-decoder controlled ARQ system on a block basis. Presumably, an upper limit to the undetected error rate could be set, and that code which is most efficient and still has an undetected error rate below the limit would be used.

The above criteria were roughly weighed along with the fact that codes of a fixed block length would be desirable; thus the Bose-Chaudhuri codes of block length 31 seemed most attractive. Again, since the error statistics are expected to vary with channel as well as with time, no particular code of this group has been selected, and each channel would vary its code appropriately and also independently of the other channels. The two low-

NRL Problems R06-06 and R06-07; Projects SF 001-08-05-9094, SS 024-001-9083, and RF 001-08-41-4552. This is an interim report; work on the problem is continuing. Manuscript submitted July 11, 1963.

est efficiency codes of block length 31 are of sufficient decoding difficulty and small enough efficiency that they would not be used. This leaves the (31,26), (31,21), and (31,16) codes. When the encoder-decoder agree that the first information bit will be a zero and hence not transmitted, these codes will now fulfill the criteria that both the number of information bits and the number of parity bits are divisible by five so the block will consist of N information characters followed by $6 - N$ parity characters for $N = 5,4,3$. Then the efficiencies will be .83, .67, .50, respectively. The encoding problem, if a general purpose encoder is not available, is relatively easy. In any cyclic code the number of parity checks is greater than or equal to the number of shift registers which in turn is greater than or equal to the number of half adders.

The literature (2,3,4,5) contains a detailed description of the nature of these codes. A few of their more important characteristics are given here along with a brief mathematical explanation. The multiple error correcting Bose-Chaudhuri codes are cyclic codes and are generalizations of the single error correcting Hamming codes. The (N,k) code of block length $N = 2^m-1$, which corrects t errors or less, has a number of parity checks $N-k \leqq mt$. Because of their cyclic nature, they can detect any burst of length $N-k$ or less and can be encoded by a shift register of length $N-k$ or k, whichever is less. The decoders, other than the single error correct, multiple error detect, are of a somewhat more complicated design (6); the single, double, and triple error correcting codes can be decoded in real time on a multichannel basis, using a small general purpose computer. Mathematically these codes are relatively easy to understand. Consider an irreducible polynomial $g(x)$ of degree $N-k$ of $GF(2)$, *i.e.*, the coefficients in the Galois Field are taken modulo 2, and $f(x) = a_{N-k}x^{N-k} + a_{N-k+1}x^{N-k+1} + \ldots + a_N x^N$. Then $f(x) = g(x)h(x) + r(x)$, where $r(x)$ is of degree $N-k-1$ or less. Since we are dealing in the coefficient field modulo 2, $f(x) - r(x) = g(x)h(x) = f(x) + r(x)$. If we wish to transmit the information bits $(a_{N-k}, a_{N-k+1}, \ldots, a_N)$, where $a_j = 1$ or 0, we need only form the polynomial $f(x)$ as above, divide this by $g(x)$ to find the remainder $r(x)$ and actually transmit the bits $(r_0, r_1, \ldots r_{N-k-1}, a_{N-k}, \ldots, a_N)$. These bits are divided into groups of five and sent as teletype characters. (Note we required previously that five will divide both N and $N-k$). Let α be a root of $g(x)$; then $g(\alpha) = 0$, and $g(\alpha)h(\alpha) = f(\alpha) + r(\alpha) = 0$. Suppose we have a single error in the *mth* bit $(m = 0,1, \ldots, 29)$. We will denote this by the polynomial $e(x) = x^m$. Let $f_1(x) + r_1(x)$ be the received polynomial, and by the single error hypothesis,

$$f_1(x) + r_1(x) = f(x) + r(x) + e(x)$$

or letting $x = \alpha$, a root of $g(x)$,

$$f_1(\alpha) + r_1(\alpha) = f(\alpha) + r(\alpha) + e(\alpha) = e(\alpha) = \alpha^m.$$

Essentially, the problem is now solved. We can calculate $f_1(\alpha) + r_1(\alpha)$ on the received message, search the table of powers of α to find the m such that $\alpha^m = f_1(\alpha) + r_1(\alpha)$, and then change the *mth* bit.

The higher order Bose-Chaudhuri codes are only slightly more difficult. Let us discuss the triple error correcting code in detail. Let

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

where if $m_1(\alpha) = 0$, then

$$m_3(\alpha^3) = m_5(\alpha^5) = 0$$

and

$$f(\alpha) + r(\alpha) = f(\alpha^3) + r(\alpha^3) = f(\alpha^5) + r(\alpha^5) = 0.$$

Again let $f_1(x) + r_1(x)$ be received where $f_1(x) + r_1(x) = f(x) + r(x) + e(x)$. Note that in general $e(x)$ is a polynomial with three nonzero coefficients. Now

$$f_1(\alpha) + r_1(\alpha) = e(\alpha) = S_1 = \alpha^j + \alpha^k + \alpha^l$$

$$f_1(\alpha^3) + r_1(\alpha^3) = e(\alpha^3) = S_3 = \alpha^{3j} + \alpha^{3k} + \alpha^{3l}$$

$$f_1(\alpha^5) + r_1(\alpha^5) = e(\alpha^5) = S_5 = \alpha^{5j} + \alpha^{5k} + \alpha^{5l}.$$

It can be verified that

$$x^3 + S_1x^2 + \frac{S_1^2S_3 + S_5}{S_1^3 + S_3}x + \left(S_1^3 + S_3 + S_1\frac{S_1^2S_3 + S_5}{S_1^3 + S_3}\right) = 0$$

has three roots which are α^j, α^k, and α^l; thus the *jth*, *kth*, and *lth* bits would be changed to correct the message. Explicit shift register encoders, computer encoding algorithms, and computer decoding algorithms are included in the Appendix

for the single, double, and triple error correcting codes.

The error detecting capability of the Bose-Chaudhuri codes is twofold. Since the Hamming distance between two code words is at least $2t+1$, then for a t error correcting code, any error of $2t$ bits can be detected. In addition, in an (N,k) cyclic code, any burst of length $N-k$ or less can be detected along with a high percentage of longer bursts. For simultaneous correction and detection, we are presented with a trade-off. Again, since the Hamming distance between two code words is $2t+1$, we may correct any C errors and detect any D errors for $C+D=2t$ and $C \leqq D$. Table 1 summarizes this fact.

TABLE 1
Correction and Detection Trade-Off

Code	No. of Corrections	No. of Error Detections
SEC	1	1
SEC*	–	2
DEC	2	2
DEC†	1	3
DEC*	–	4
TEC	3	3
TEC‡	2	4
TEC†	1	5
TEC*	–	6

*Used only for error detection.
†Used to correct single errors at most.
‡Used to correct double errors at most.

There is also a limited amount of detection capability from two other sources. In order to shorten the block length from 31 bits to 30 bits, both transmitter and receiver have agreed that $a_{30}=0$; if the algorithm concludes that a_{30} is in error, we have actually detected an error which is beyond the correction capability of the code. In addition, a polynomial of degree N must have N distinct (except for the zero solution) solutions; any other cases are caused by error beyond the correcting capability of the code.

SUMMARY

The Bose-Chaudhuri codes were selected for their high efficiency and elementary nature. A class of codes of the same block length provides great versatility at a very small increase in program size, which in turn is due to the large amount of common information. The expected message length and real time requirement lead us to believe that the code of block length 31 would be best. Finally, a computer controlled ARQ system on a block basis would be highly efficient and would lower the net error rate to a negligible value.

REFERENCES

1. Fontaine, A.B., "Error Statistics and Coding for Digital Data Transmission over Telephone and Teletype Circuits," *Proc. IEEE* **51**:474-475 (Mar. 1963)
2. Bartee, T.C., and Schneider, D.I., "An Electronic Decoder for Bose-Chaudhuri – Hocquenghem Error – Correcting Codes," *IRE Transactions on Information Theory*, Vol. IT-8, pp. S17-S24, Sept. 1962
3. Bose, R.C., and Ray-Chaudhuri, D.K., "On a Class of Error-Correcting Binary Group Codes," *Information and Control* **3**:68-79 (Mar. 1960)
4. Peterson, W.W., "Encoding and Error-Correction Procedures for the Bose-Chaudhuri Codes," *IRE Transactions on Information Theory*, Vol. IT-6, pp. 459-470, Sept. 1960
5. Peterson, W.W., "Error Correcting Codes," The M.I.T. Press, pp. 162-182, 1961
6. Perry, K.E., and Wozencraft, J.M., "SECO: A Self-Regulating Error Correcting Coder-Decoder," *IRE Transactions on Information Theory*, Vol. IT-8, pp. S128-S135, Sept. 1962

Appendix A
Implementation of Bose-Chaudhuri Codes on a Digital Computer

The irreducible polynomials of degree 5 are:

$$m_1(x) = 1 + x^2 + x^5$$
$$m_3(x) = 1 + x^2 + x^3 + x^4 + x^5$$
$$m_5(x) = 1 + x + x^2 + x^4 + x^5.$$

Note that if α is a root of $m_1(x)$, then α^3 is a root of $m_3(x)$, and α^5 is a root of $m_5(x)$. The parity check matrix is formed by powers of α modulo α^{31}.

$$M = \begin{vmatrix} \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^1 & \alpha^3 & \alpha^5 \\ \alpha^2 & \alpha^6 & \alpha^{10} \\ \alpha^3 & \alpha^9 & \alpha^{15} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \alpha^{30} & \alpha^{90} = \alpha^{28} & \alpha^{150} = \alpha^{26} \end{vmatrix}$$

If we chose $\alpha^0 = \alpha^{31} = (10000)$, then the M matrix is explicitly:

$$M = \left|\begin{array}{ccc}
10000 & 10000 & 10000 \\
01000 & 00010 & 10100 \\
00100 & 01010 & 10001 \\
00010 & 01011 & 11111 \\
00001 & 01110 & 00110 \\
10100 & 11111 & 10011 \\
01010 & 11000 & 01001 \\
00101 & 00011 & 00001 \\
10110 & 01111 & 01011 \\
01011 & 11010 & 10111 \\
10001 & 01001 & 01100 \\
11100 & 00100 & 01111 \\
01110 & 10100 & 10010 \\
00111 & 10110 & 00010 \\
10111 & 11100 & 10110 \\
11111 & 10111 & 00111 \\
11011 & 11001 & 11000 \\
11001 & 00110 & 11110 \\
11000 & 11110 & 01101 \\
01100 & 11101 & 00100 \\
00110 & 10010 & 00101 \\
00011 & 01000 & 01110 \\
10101 & 00001 & 11001 \\
11110 & 00101 & 10101 \\
01111 & 10001 & 11010 \\
10011 & 00111 & 01000 \\
11101 & 11011 & 01010 \\
11010 & 01100 & 11100 \\
01101 & 10101 & 11011 \\
10010 & 10011 & 00011 \\
01001 & 01101 & 11101
\end{array}\right|$$

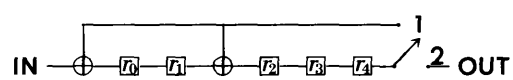
If $R(x)$ is the received function and $R_{30} = 0$ as was previously defined, then $(R_0, R_1, \ldots, R_{30})\, M = (S_1, S_3, S_5)$, where $S_1 = R(\alpha)$, $S_3 = R(\alpha^3)$, and $S_5 = R(\alpha^5)$ are each five bits. All additions are modulo two, *i.e.*, no carry; for instance, $\alpha^8 + \alpha^{14} = (10110) + (10111) = (00001) = \alpha^4$. All multiplications and divisions are in Galois Field code; as, for example, $\alpha^{19}\alpha^{24} = (01100)(01111) = \alpha^{43} = \alpha^{43-31} = \alpha^{12} = (01110)$. Alternatively, one can consider this as polynomial multiplication modulo an irreducible polynomial $(1 + x^2 + x^5)$ whose coefficients are taken modulo two. For example,

$$\begin{array}{l}
\alpha^{19} = 01100 \\
\alpha^{24} = 01111 \\
\hline
\quad 001100 \\
\quad\;\; 01100 \\
\quad\quad\;\; 01100 \\
\quad\quad\quad\;\; 01100 \\
\hline
\end{array}$$

$$\begin{aligned}
001000100 &= (00100) + 0\alpha^5 + 1\alpha^6 + 0\alpha^7 + 0\alpha^8 \\
&= (00100) + (01010) \\
&= (01110) = \alpha^{12}.
\end{aligned}$$

SINGLE ERROR CORRECTING CODE

For the single error correcting code, the generator polynomial is $g(x) = m_1(x) = 1 + x^2 + x^5$. This polynomial describes the encoder,



where the feedback connectors from the fifth stage go to the zeroth and second stage. To encode, first reset all the registers to zero, put the switch in position 1, feed the data bits into the shift register and simultaneously to the output line. After 25 bits (the five data characters) have been transmitted, put the switch in position 2 and shift out the last 5 bits (sixth character) to the transmitter. What we are actually doing is dividing the 25-bit data polynomial $f(x) = a_5x^5 + a_6x^6 + \ldots + a_{29}x^{29}$ by the generator polynomial $g(x) = 1 + x^2 + x^5$ to get a remainder $r(x) = r_0 + r_1x + r_2x^2 + r_3x^3 + r_4x^4$, which in turn is added to the data to form the total transmitted polynomial $f(x) + r(x) = r_0 + \ldots + r_4 + a_4x^4 + a_5x^5 + \ldots + a_{29}x^{29}$. Note that a_{30} is assumed equal to zero and not transmitted.

If a general purpose encoder is available, the encoding can be done identically in a more efficient manner. Each of the bits r_0, r_1, r_2, r_3, and r_4 is simply the parity of the sum of certain information bits listed below:

r_0 = Parity (5,8,10,11,14,15,16,17,18,22,23,25,26,27,29)

r_1 = Parity (6,9,11,12,15,16,17,18,19,23,24,26,27,28,30)

r_2 = Parity (5,7,8,11,12,13,14,15,19,20,22,23,24,26,28)

r_3 = Parity (6,8,9,12,13,14,15,16,20,21,23,24,25,27,29)

r_4 = Parity (7,9,10,13,14,15,16,17,21,22,24,25,26,28,30).

Since this is a single error correcting code, either the Hamming shift register decoder or the more general Bose-Chaudhuri decoder can be used. We will discuss only the latter method. The received polynomial $R(x)$ in row matrix form is multiplied by the first five columns of M to yield S_1. If $S_1 = 0$, no error occurred; if not, there exists an $\alpha^k = S_1$. If $k \neq 30$, a single error has occurred in the *kth* bit; otherwise a multiple error has occurred.

DOUBLE ERROR CORRECTING CODE

The generator polynomial for the double error correcting code is $g(x) = m_1(x)m_3(x) = 1 + x^3 + x^5 + x^6 + x^8 + x^9 + x^{10}$. The shift register encoder has ten stages, where the feedback from the tenth stage goes to the stages 0, 3, 5, 6, 8, and 9. The encoding procedure is identical with the single error correcting code with the exception that there are only 20 data bits (a_{10} through a_{29}) and 10 parity check bits (r_0 through r_9).

The general purpose encoder is given by the following parity equations:

r_0 = Parity (10,11,13,15,17,18,19,20,23,26,28)
r_1 = Parity (11,12,14,16,18,19,20,24,27,29)
r_2 = Parity (12,13,15,17,19,20,21,22,25,28,30)
r_3 = Parity (10,11,14,15,16,17,19,21,22,28,29)
r_4 = Parity (11,12,15,16,17,18,20,22,23,29,30)
r_5 = Parity (10,11,12,15,16,20,21,24,26,28,30)
r_6 = Parity (10,12,15,16,18,19,20,21,22,23,25,26,27,28,29)
r_7 = Parity (11,13,16,17,19,20,21,22,23,24,26,27,28,29,30)
r_8 = Parity (10,11,12,13,14,15,19,21,22,24,25,26,27,29,30)
r_9 = Parity (10,12,14,16,17,18,19,22,25,27,30).

The decoding procedure is similar to the single error correcting code. The received function $R(x)$ is multiplied by the first 10 columns of M. The first five bits of the product will be $S_1 = R(\alpha)$, and the second five bits will be $S_3 = R(\alpha^3)$. An error in the *kth* bit will result in α^k being a solution of the equation:

$$x^2 + S_1 x + \frac{S_1^3 + S_3}{S_1} = 0.$$

If no error occurred, then $S_1 = S_3 = 0$. If only one error occurred, then $S_1 = \alpha^k$, $S_3 = \alpha^{3k}$, $S_1^3 + S_3 = 0$; and the equation in x given here can be solved by the single error correcting subroutine. Errors of more than two bits can be detected in three ways: first, if $S_1 = 0$ and $S_3 \neq 0$; second, if $S_1 \neq 0$ and $S_3 = 0$; third, if α^{30} is a root of the above polynomial. It is of interest to note that if two errors did occur, say in the *nth* and *mth* bits, then $S_1 = \alpha^m + \alpha^N$, $S_3 = \alpha^{3m} + \alpha^{3N}$, and the polynomial becomes $x^2 + (\alpha^m + \alpha^N)x + \alpha^{m+N} = 0$, which obviously has solutions α^m and α^N as predicted.

The normal method of error correction would be to substitute $x = \alpha^k$ for $k = 0,1, \ldots, 29$ into this polynomial. For some $k = m$, $x = \alpha^m$ is a solution, and $\alpha^N = S_1 + \alpha^m$ is also. If the errors are random, one must perform about nine trial solutions before the first solution is found. A much faster method exists.* If we note that

$$1 + \frac{S_3}{S_1^3} = \frac{\alpha^{m-N}}{1 + \alpha^{2(m-N)}} = \frac{\alpha^{N-m}}{1 + \alpha^{2(N-m)}}$$

is solely a function of $|m-N|$ and remember the cyclic nature of the code, we may say $|m-N| \leq 15$. Therefore, there are only 15 distinct values of $1 + S_3/S_1^3$, not including zero, and corresponding values of α^{m-N}. With this information we may calculate $\alpha^N = (S_1/1 + \alpha^{m-N})$ and $\alpha^m = S_1 + \alpha^N$. The following tabulation gives the distinct values of $|m-N|$ and the corresponding values of $1 + S_3/S_1^3$.

$\lvert m-N \rvert$	$1 + S_3/S_1^3$
1	$\alpha^{27} = (1\ 1\ 0\ 1\ 0)$
2	$\alpha^{23} = (1\ 1\ 1\ 1\ 0)$
3	$\alpha^{7} = (0\ 0\ 1\ 0\ 1)$
4	$\alpha^{15} = (1\ 1\ 1\ 1\ 1)$
5	$\alpha^{1} = (0\ 1\ 0\ 0\ 0)$
6	$\alpha^{14} = (1\ 0\ 1\ 1\ 1)$
7	$\alpha^{25} = (1\ 0\ 0\ 1\ 1)$
8	$\alpha^{30} = (0\ 1\ 0\ 0\ 1)$
9	$\alpha^{8} = (1\ 0\ 1\ 1\ 0)$
10	$\alpha^{2} = (0\ 0\ 1\ 0\ 0)$
11	$\alpha^{4} = (0\ 0\ 0\ 0\ 1)$
12	$\alpha^{28} = (0\ 1\ 1\ 0\ 1)$
13	$\alpha^{16} = (1\ 1\ 0\ 1\ 1)$
14	$\alpha^{19} = (0\ 1\ 1\ 0\ 0)$
15	$\alpha^{29} = (1\ 0\ 0\ 1\ 0)$

*R. B. Banerji, "A Decoding Procedure for Double Error Correcting Bose-Ray-Chaudhuri Codes," *Proceedings of the IRE*, **49**:1585 (Oct. 1961)

TRIPLE ERROR CORRECTING CODE

The generator polynomial for the triple error correcting code is $g(x) = m_1(x)m_3(x)m_5(x) = 1 + x + x^2 + x^3 + x^5 + x^7 + x^8 + x^9 + x^{10} + x^{11} + x^{15}$. The shift register encoder has a feedback connection from the 15th stage to the stages 0, 1, 2, 3, 5, 7, 8, 9, 10, and 11.

The encoding procedure is identical with the single and double error correcting codes with the exception that there are 15 information bits and 15 parity bits. The general purpose encoder is given by the following parity equations:

r_0 = Parity (15,19,20,21,22,27,30)
r_1 = Parity (15,16,19,23,27,28,30)
r_2 = Parity (15,16,17,19,21,22,24,27,28,29,30)
r_3 = Parity (15,16,17,18,19,21,23,25,27,28,29)
r_4 = Parity (16,17,18,19,20,22,24,26,28,29,30)
r_5 = Parity (15,17,18,22,23,25,29)
r_6 = Parity (16,18,19,23,24,26,30)
r_7 = Parity (15,17,21,22,24,25,30)
r_8 = Parity (15,16,18,19,20,21,23,25,26,27,30)
r_9 = Parity (15,16,17,24,26,28,30)
r_{10} = Parity (15,16,17,18,19,20,21,22,25,29,30)
r_{11} = Parity (15,16,17,18,23,26,27)
r_{12} = Parity (16,17,18,19,24,27,28)
r_{13} = Parity (17,18,19,20,25,28,29)
r_{14} = Parity (18,19,20,21,26,29,30)

The decoding procedure is to multiply the received function $R(x)$ by all 15 columns of M. The first five bits are S_1, the second are S_3, and the third are S_5. An error in the kth bit will cause $x = \alpha^k$ to be a solution of the polynomial

$$x^3 + S_1x^2 + \frac{S_1^2S_3 + S_5}{S_1^3 + S_3}x + \left(S_1^3 + S_3 + \frac{S_1\,(S_1^2S_3 + S_5)}{S_1^3 + S_3}\right) = 0.$$

If $S_1 = S_3 = S_5 = 0$, there was no error. For a single error, $S_1^3 + S_3 = 0$ and $S_1^2S_3 + S_5 = 0$. A double error will cause the constant term in the above polynomial to be zero. An uncorrectable error has been detected (a) if $S_1^2 + S_3 = 0$ but $S_1^2S_3 + S_5 \neq 0$, (b) if α^{30} is a root, or (c) if three distinct roots to the above polynomial do not exist.

When three errors do occur, if the trial solutions α^N are selected randomly, one would expect to try about eight trials before the first solution is found. If the errors occur in the jth, kth, and lth positions, then $S_1 = \alpha^j + \alpha^k + \alpha^l$; and if the first bit of S_1 is zero, either one or all three of the solutions begin with a zero. With this information we would try only those trial solutions α^N whose first bit is a zero, thereby cutting down the expected number of trial solutions to six trials before the first solution is found. Obviously, these same statements hold if the word zero is replaced with the word one. After one solution, say α^j, has been found, we may divide the third degree polynomial by $x + \alpha^j$ to yield a second degree polynomial which can be solved by the double error correcting procedure. This can be done most efficiently by simply substituting for the terms S_1 and S_3 in the double error correcting program the values $S_1 + \alpha^j$ and $S_3 + \alpha^{3j}$, respectively.

NUMERICAL EXAMPLE OF TRIPLE ERROR CORRECTING ENCODING AND DECODING

Suppose we wish to transmit the three letters TEC; in teletype T = (00001), E = (10000), C = (01110). The T would enter the encoder first, so to agree with the notation for $f(x)$: f = 01110 10000 00001, where the ones and zeros can be thought of as coefficients of the polynomial $f(x) = 0x^{15} + 1x^{16} + 1x^{17} + 1x^{18} + \ldots + 1x^{29}$. From the TEC encoding algorithm $r_0 = 1$, $r_1 = 1$, $r_2 = 1$, $r_3 = 0$, $r_4 = 1$, $r_5 = 1$, $r_6 = 0$, $r_7 = 1$, $r_8 = 1$, $r_9 = 0$, $r_{10} = 1$, $r_{11} = 1$, $r_{12} = 1$, $r_{13} = 0$, $r_{14} = 1$. So now the transmitted message would be $f(x) + r(x)$, 11101 10110 11101 01110 10000 00001, where the right-hand 5 bits would be transmitted first. Suppose an error occurred in the 10th, 17th, and 22nd bits. Remembering that the first bit at the left is bit zero, one sees that the received message would be $R(x)$ = 11101 10110 01101 01010 10100 00001. Again, since both transmitter and receiver agreed that bit thirty would be zero, this bit is now added to the right-hand side of $R(x)$ to make a 1 by 31 matrix which is multiplied on the right by the 31 by 15 M-matrix to yield a 1 by 15 matrix (11101 01110 01011). Then

$$S_1 = 11101 = \alpha^{26}$$

$$S_3 = 01110 = \alpha^{12}$$

$$S_5 = 01011 = \alpha^{9}$$

$$S_1^3 = \alpha^{78} = \alpha^{78-62} = \alpha^{16} = 11011$$

$$S_1^3 + S_3 = 11011 + 01110 = 10101 = \alpha^{22} \neq 0$$

$$S_1^2 S_3 = \alpha^{52}\ \alpha^{12} = \alpha^{64} = \alpha^2 = 00100$$

$$S_1^2 S_3 + S_5 = 00100 + 01011 = 01111 = \alpha^{24} \neq 0$$

$$\frac{S_1(S_1^2 S_3 + S_5)}{S_1^3 + S_3} = \frac{\alpha^{26}\ \alpha^{24}}{\alpha^{22}} = \alpha^{28} = 01101$$

$$S_1^3 + S_3 + \frac{S_1(S_1^2 S_3 + S_5)}{S_1^3 + S_3} = 10101 + 01101 = 11000 = \alpha^{18} \neq 0.$$

So we have at least a triple error

$$\frac{S_1^2 S_3 + S_5}{S_1^3 + S_3} = \frac{\alpha^{24}}{\alpha^{22}} = \alpha^2.$$

Then the polynomial is

$$x^3 + \alpha^{26}x^2 + \alpha^2 x + \alpha^{18} = 0.$$

Since the first bit of S_1 is a one, we try only those solutions α^n whose first bit is a one, namely: α^0, α^5, α^8, α^{10}, etc. $x = \alpha^{10}$ will be a solution since $\alpha^{30} + \alpha^{26}$ $\alpha^{20} + \alpha^2\alpha^{10} + \alpha^{18} = \alpha^{30} + \alpha^{15} + \alpha^{12} + \alpha^{18} = 01001 + 11111 + 01110 + 11000 = 00000$.

We then substitute for the values of S_1 and S_3 in the double error correcting program the values $S_1 + \alpha^{10} = 11101 + 10001 = 01100 = \alpha^{19}$ and $S_3 + \alpha^{30} = 01110 + 01001 = 00111 = \alpha^{13}$ respectively. Now following the procedure for the double error correcting program with the new values of S_1 and S_3,

$$1 + \frac{S_3}{S_1^3} = \alpha^0 + \frac{\alpha^{13}}{\alpha^{57}} = \alpha^0 + \alpha^{62+13-57} = \alpha^0 + \alpha^{18} = 10000 + 11000 = 01000 = \alpha^1.$$

From the table of $|m-N|$ and $1 + S_3/S_1^3$, we see that

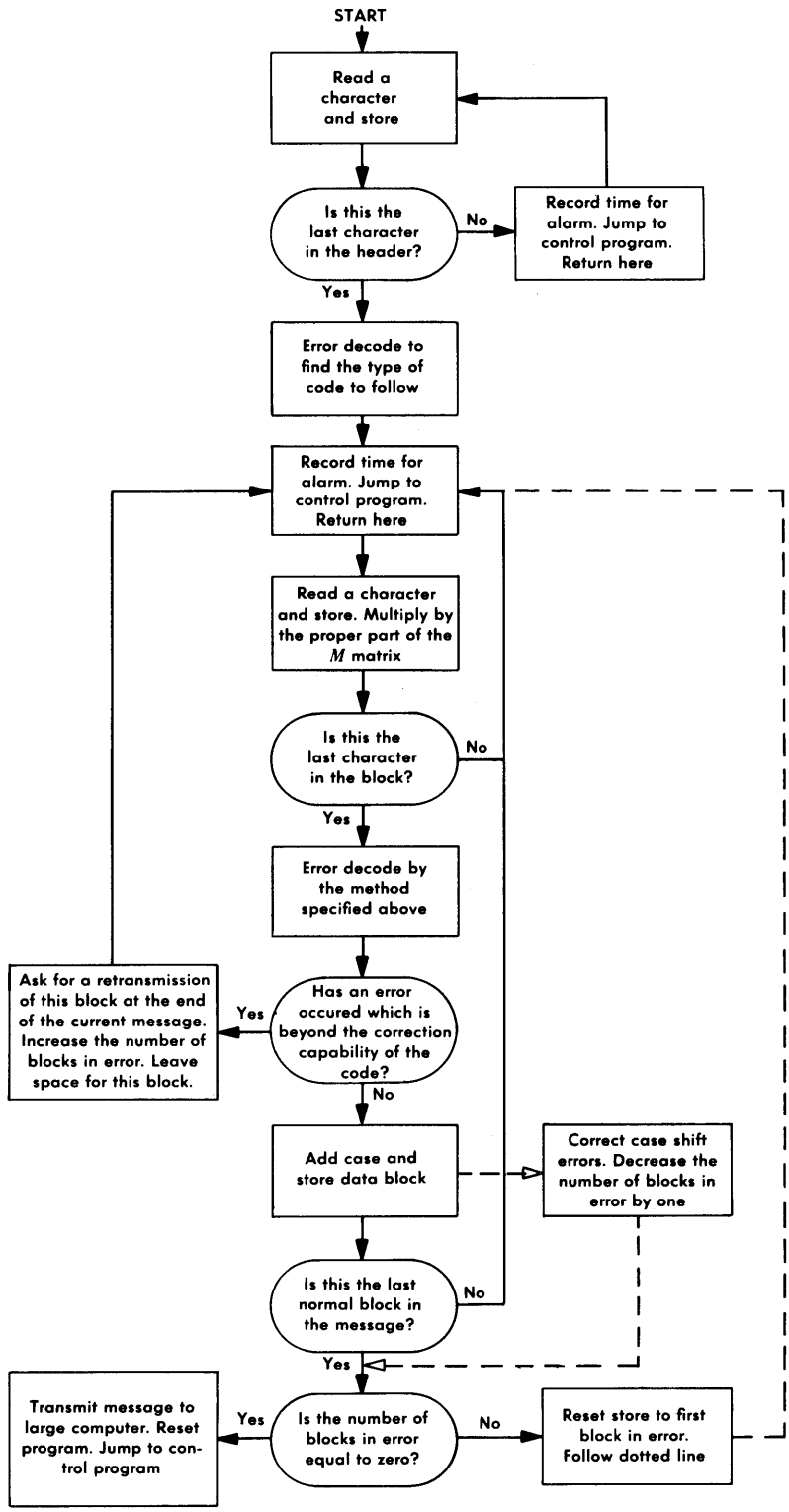
$$\alpha^N = \frac{S_1}{1 + \alpha^{m-N}} = \frac{\alpha^{19}}{\alpha^0 + \alpha^5} = \frac{\alpha^{19}}{(10000) + (10100)} = \frac{\alpha^{19}}{00100} = \frac{\alpha^{19}}{\alpha^2} = \alpha^{17}$$

and

$$\alpha^m = S_1 + \alpha^{17} = 01100 + 11001 = 10101 = \alpha^{22}.$$

The three solutions of the triple error correcting polynomial are $x = \alpha^{10}$, α^{17}, α^{22}. The errors are in the 10th, 17th, and 22nd bits; and these bits would be changed.

COMPUTER FLOW DIAGRAMS



Subroutine for Partial Matrix Product *RM*

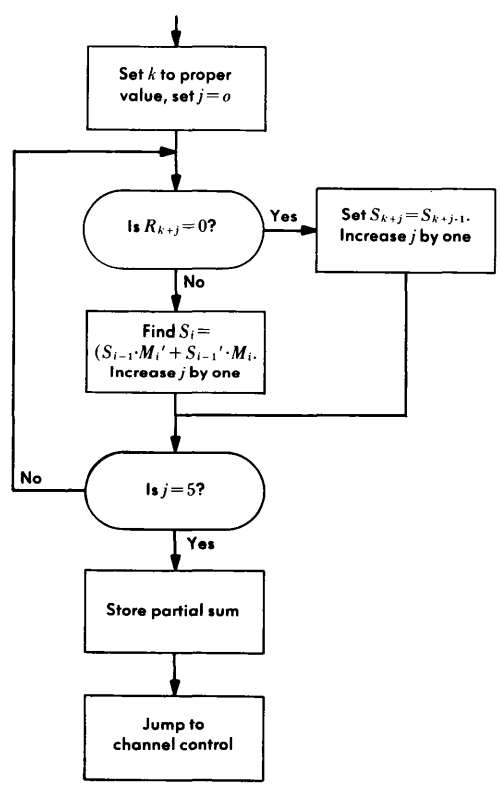
The equation for this routine is

$$S_i = (S_{i-1} \cdot M'_i + S'_{i-1} \cdot M_i)\ \delta_{Ri,1} + S_{i-1}\delta_{Ri,0}$$

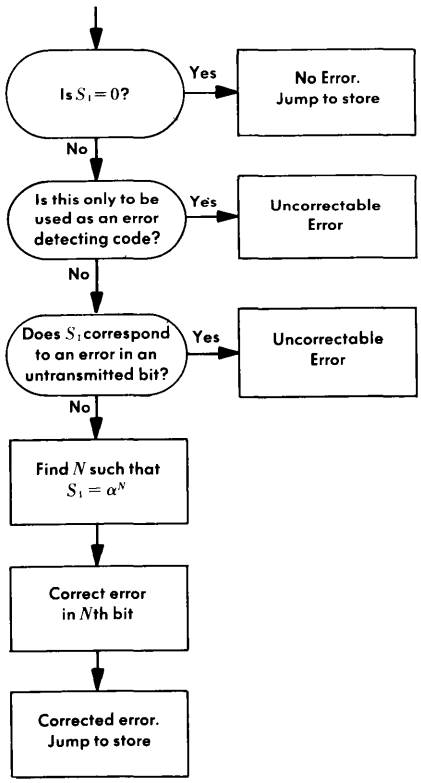
where $i = 5k,\ 5k+1,\ \ldots,\ 5k+4$; and for a block length of 30, $k = 5, 4, \ldots, 0$ for the first, second, . . ., sixth input characters, respectively. Here A' equals the ones compliment of A, and $A \cdot B$ is a logical end, and

$$\delta_{Ri,1} = \begin{cases} 1 & \text{if } R_i = 1 \\ 0 & \text{if } R_i = 0 \end{cases}$$

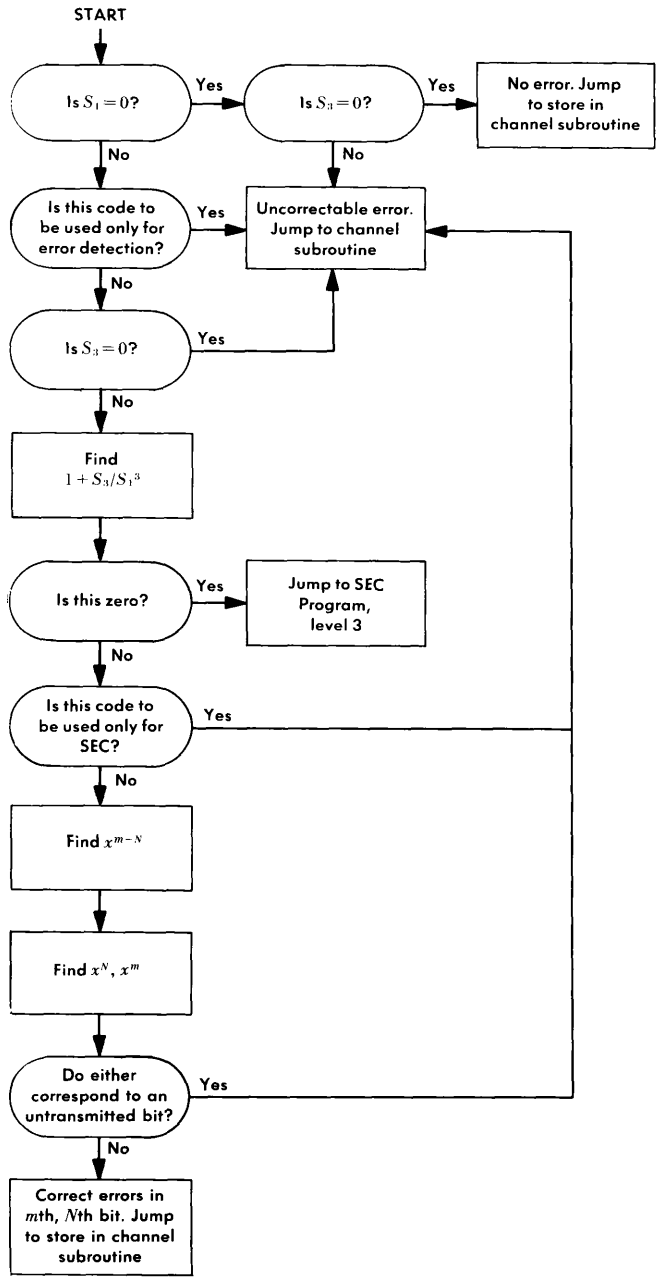
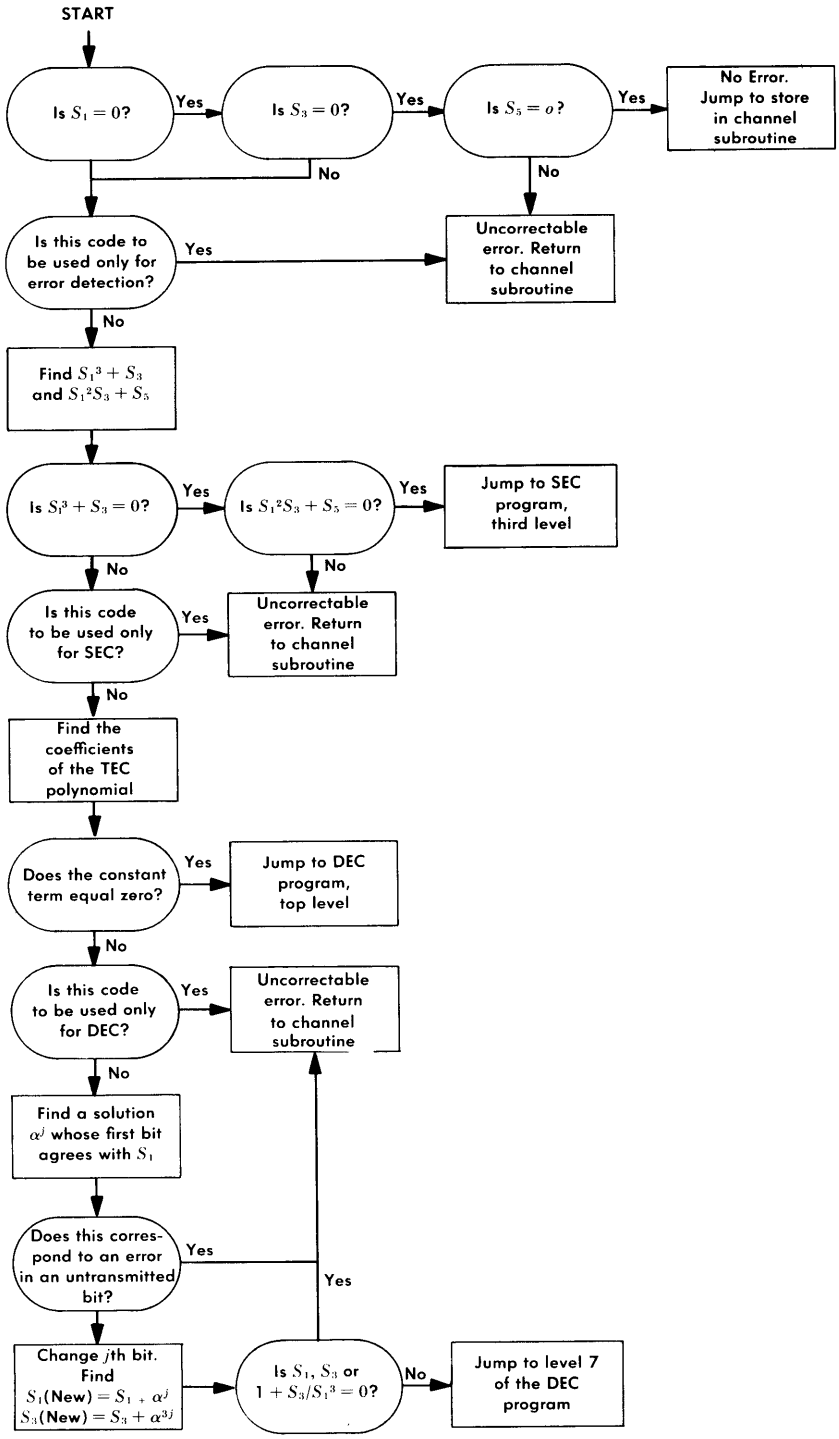
$$\delta_{Ri,0} = \begin{cases} 0 & \text{if } R_i = 1 \\ 1 & \text{if } R_i = 0. \end{cases}$$



Single Error Correcting Code with *RM* Premultiplied



Double Error Correcting Code with *RM* Premultiplied



Triple Error Correcting Program with *RM* Premultiplied