3-1972

Decoding by Sequential Code Reduction

Luther D. Rudolph
Syracuse University

Carlos R.P. Hartmann
Syracuse University, chartman@syr.edu

72-3

DECODING BY SEQUENTIAL CODE REDUCTION

LUTHER D. RUDOLPH
CARLOS R. P. HARTMANN

MARCH, 1972

SYSTEMS AND INFORMATION SCIENCE
SYRACUSE UNIVERSITY

DECODING BY SEQUENTIAL CODE REDUCTION

LUTHER D. RUDOLPH

CARLOS R.P. HARTMANN

This work was supported by the

Rome Air Development Center under Contracts

F 30602-70-C-0060 and
F 30602-70-C-0190

SYSTEMS AND INFORMATION SCIENCE
SYRACUSE UNIVERSITY

SYRACUSE, NEW YORK 13210
(315) 476-5541 Ext. 2368

ABSTRACT

A general decoding method for cyclic codes is presented which gives promise of substantially reducing the complexity of decoders at the cost of a modest increase in decoding time (or delay). Significant reductions in decoder complexity for binary cyclic finite-geometry codes are demonstrated, and two decoding options for the Golay code are presented.

ACKNOWLEDGEMENT

The authors wish to acknowledge the contributions to this study by Fred D. Schmandt of the Rome Air Development Center and Ralph J. Longobardi, a graduate student at Syracuse University.

TABLE OF CONTENTS

SECTION I

INTRODUCTION

The major problem in the practical application of error-correcting codes is the complexity of decoding. One of the basic decoding complexity trade-offs is that between decoding time and hardware cost. Decoding algorithms range from expensive combinatorial schemes that operate in minimum time to slow sequential schemes that require a minimal amount of hardware. Surprisingly, there seem to be few decoding schemes in the middle range where one would expect to find the economical operating points. Consideration of this time-hardware trade-off has led to the discovery of a new decoding technique -- decoding by sequential code reduction -- that gives promise of significantly reducing the complexity of combinatorial decoders at the cost of a modest increase in decoding time (or delay). This scheme applies to all cyclic codes and perhaps to other codes as well. Significant reductions in decoder complexity have already been demonstrated for a number of cyclic codes.

In Section 2, a general bounded-distance decoding algorithm for cyclic codes is formulated and sequential

code reduction is introduced by means of an example. The application of sequential code reduction to finite-geometry codes is considered in Section 3, and some preliminary results are given. In Section 4, some implementation options for the Golay (23,12) code are presented. Section 5 contains a discussion of the results.

We have assumed that the reader is familiar with coding theory at the level of, say, W. W. Peterson and E. J. Weldon, Jr.'s "Error-Correcting Codes", 2nd Edition, (M.I.T. Press, 1972). For the sake of simplicity, only binary cyclic codes will be considered in this report.

SECTION 2

SEQUENTIAL CODE REDUCTION

In this section we introduce the concept of sequential code reduction by means of an example. The standard bounded-distance decoding algorithm for cyclic codes is formulated and then applied to the (7,4) Hamming code in the form of a conventional 2-step majority logic decoding algorithm. It is then shown how this combinatorial algorithm can be converted to a sequential code reduction algorithm at the cost of an increase in decoding time. This is followed by a brief discussion of the basic ideas involved in sequential code reduction.

2.1 Bounded-distance decoding of cyclic codes

A bounded-distance decoding algorithm for an (n,k) t-error-correcting code is guaranteed to correct all errors of weight t or less. A few bounded-distance decoding algorithms (e.g. majority logic decoding) correct some patterns of more than t errors, but this excess correction capability is due to accident rather than design and can be considered a bonus obtained at no extra cost.

Let $c = (c_0,\ldots,c_{n-1})$ be the transmitted code word, $e = (e_0,\ldots,e_{n-1})$ the error vector and $c + e = r = (r_0,\ldots,r_{n-1})$ the received word, where '+' denotes vector addition over GF(2). (All arithmetic in this report is over GF(2) unless otherwise stated.) Let H be a reduced parity check matrix for the code and $\hat{e} = (\hat{e}_0,\ldots,\hat{e}_{n-1})$ and $\hat{c} = (\hat{c}_0,\ldots,\hat{c}_{n-1})$ the decoder's estimate of e and c respectively. $W_H(x)$ will denote the Hamming weight of x. Since we are considering cyclic codes only, the decoding algorithm need only be capable of correctly determining $\hat{c}_0$ whenever t or fewer errors have occurred. A general algorithm to do this is the following:

General Bounded-Distance Decoding Algorithm

1. Calculate the syndrome $s = Hr$
2. Solve for $\hat{e}_0$ in
$$H\hat{e} = s$$
$$W_H(\hat{e}) \leq t$$
3. Set $\hat{c}_0 = r_0 + \hat{e}_0$.

Step 2 of this algorithm may be viewed in the following way. There are 2^k solutions to the linear matrix equation $H\hat{e} = s$. The effect of the nonlinear constraint $W_H(e) \leq t$ is to reduce the number of solutions from 2^k to exactly 1 (under the assumption that $W_H(\hat{e}) \leq t$). This reduction is traditionally accomplished in one step

using a nonlinear (over GF(2)) combinational logic circuit. However -- and this is the basis for decoding by sequential code reduction -- there is no reason why this can't be accomplished sequentially, in stages.

This basic idea can perhaps best be illustrated by means of an example. We will consider the decoding of the (7,4) single-error-correcting Hamming code, first by a conventional 2-step majority decoding algorithm and then by 2-stage sequential code reduction.

2.2 Decoding algorithms for the (7,4) code

The matrix

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

is a reduced parity check matrix for the (7,4) code. The first step of the general bounded-distance decoding algorithm given above is to calculate the syndrome

$$\underset{s}{\begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix}} = \underset{H}{\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}} \underset{r}{\begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \end{bmatrix}}$$

The second step is to solve for $\hat{e}_0$ in the equation

$$\underset{H}{\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}} \underset{\hat{e}}{\begin{bmatrix} \hat{e}_0 \\ \hat{e}_1 \\ \hat{e}_2 \\ \hat{e}_3 \\ \hat{e}_4 \\ \hat{e}_5 \\ \hat{e}_6 \end{bmatrix}} = \underset{s}{\begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix}}$$

subject to the constraint

$$W_H(\hat{e}) \leq 1.$$

The number of solutions for $\hat{e}$ in the unconstrained linear equation is $2^k = 2^4 = 16$. The problem is to find the one solution for $\hat{e}$ that also satisfies the nonlinear constraint. In majority logic decoding, this is accomplished by deriving new parity checks from the old (the syndrome) using the nonlinear majority function. These new parity checks are valid only under the assumption that $W_H(e) \leq t$. The effect of adding these parity checks to the syndrome, and the corresponding rows to the parity check matrix H, is to increase the rank of H and thereby decrease the number of solutions for $\hat{e}$. New parity checks are added until the rank of H is n, in which case there is a unique solution for $\hat{e}$.

(Actually, as we shall see in this example, it is only necessary to add parity checks until the solutions for $\hat{e}$ all have the same value for $\hat{e}_0$.)

For example, consider the new parity check $s_4 = \text{maj}\{0,s_1,s_2\}$. Since s_1 and s_2 are orthogonal on $\hat{e}_0$ and $\hat{e}_1$, s_4 will give the correct value of $e_0 + e_1$ if $W_H(\hat{e}) \leq 1$, i.e.

$$s_4 = \text{maj}\{0,s_1,s_2\} = \widehat{e_0 + e_1}$$

which we can think of as the product

$$\begin{bmatrix}1 & 1 & 0 & 0 & 0 & 0 & 0\end{bmatrix} \begin{bmatrix}\hat{e}_0\\ \hat{e}_1\\ \hat{e}_2\\ \hat{e}_3\\ \hat{e}_4\\ \hat{e}_5\\ \hat{e}_6\end{bmatrix} = \begin{bmatrix}s_4\end{bmatrix}.$$

Adding this new equation to the original set gives

$$\begin{bmatrix}1 & 1 & 1 & 0 & 1 & 0 & 0\\ 1 & 1 & 0 & 1 & 0 & 0 & 1\\ 1 & 0 & 1 & 0 & 0 & 1 & 1\\ 1 & 1 & 0 & 0 & 0 & 0 & 0\end{bmatrix} \begin{bmatrix}\hat{e}_0\\ \hat{e}_1\\ \hat{e}_2\\ \hat{e}_3\\ \hat{e}_4\\ \hat{e}_5\\ \hat{e}_6\end{bmatrix} = \begin{bmatrix}s_1\\ s_2\\ s_3\\ s_4\end{bmatrix}.$$

Since the addition of the new row has increased the rank of H from 3 to 4, the number of solutions for $\hat{e}$ has been reduced from $2^4 = 16$ to $2^3 = 8$. In similar manner, we can define two other new parity checks

$$s_5 = \text{maj}\{0, s_1, s_3\} = \widehat{e_0 + e_2}$$

$$s_6 = \text{maj}\{0, s_4, s_5\} = \hat{e}_0.$$

Adding these two new parity checks further extends the decoding equation to

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{e}_0 \\ \hat{e}_1 \\ \hat{e}_2 \\ \hat{e}_3 \\ \hat{e}_4 \\ \hat{e}_5 \\ \hat{e}_6 \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \end{bmatrix}$$

The rank of H is now 6, so there are $2^1 = 2$ solutions for $\hat{e}$. However, both solutions have $\hat{e}_0 = s_6$ and the process of reducing the solution space by the addition of new parity checks may be terminated.

The final step of the algorithm is to obtain $\hat{c}_0$ from

$$\hat{c}_0 = r_0 + \hat{e}_0 = r_0 + s_6$$

The corresponding conventional 2-step majority logic decoder is shown in Figure 1.

Let s_4, s_4', s_4'', etc., denote the sequence of outputs from the upper left majority gate that results when the received word r is ring-shifted in the buffer. Since $s_4 = \widehat{e_0 + e_1}$ and the code is cyclic, we have

$$s_4 = \widehat{e_0 + e_1}$$

$$s_4' = \widehat{e_1 + e_2}$$

$$s_4'' = \widehat{e_2 + e_3}$$

$$\vdots$$

$$s_4^{VI} = \widehat{e_6 + e_0}$$

But note that

$$s_4 + s_4' = \widehat{(e_0 + e_1)} + \widehat{(e_i + e_2)} = \widehat{e_0 + e_2} = s_5.$$

It is therefore not necessary to have a separate majority gate to calculate s_5 if we are willing to store s_4, s_4', s_4'', etc., and implement the equation

$$s_5 = s_4 + s_4'.$$

The resulting 2-stage sequential code reduction decoding circuit is shown in Figure 2.

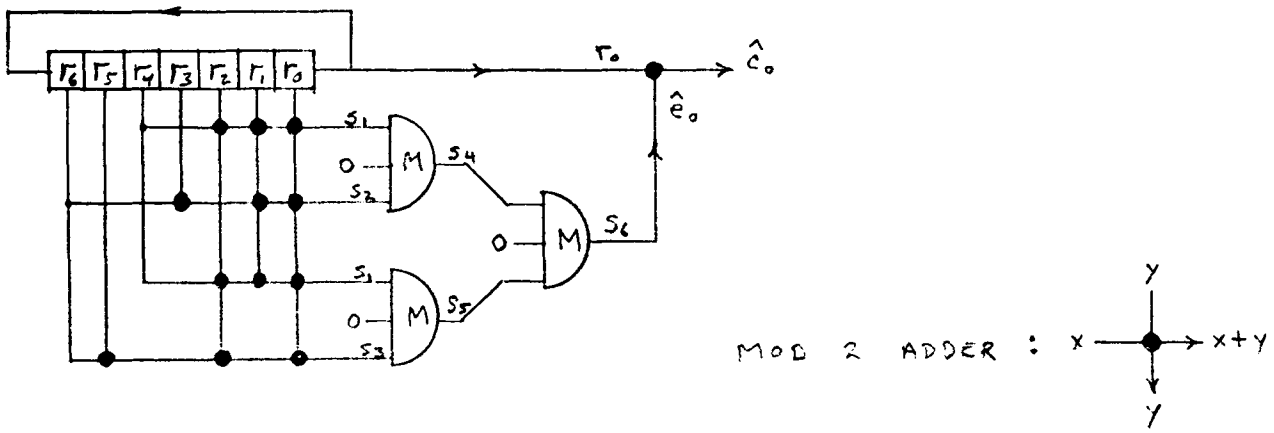


Figure 1. Conventional 2-step Majority Decoding Circuit

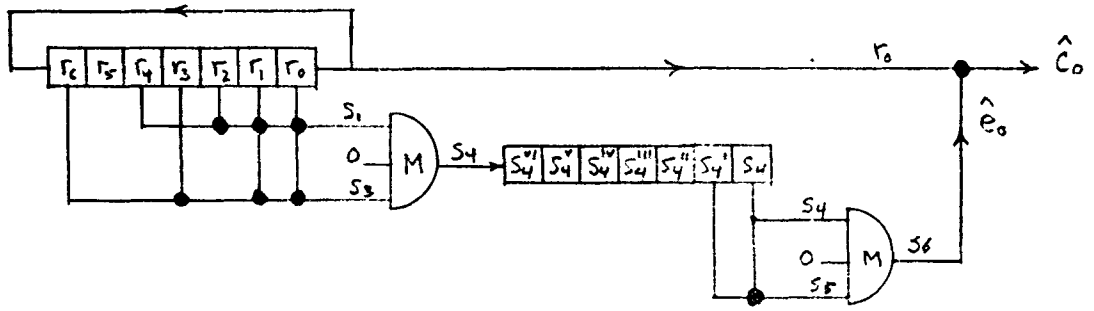
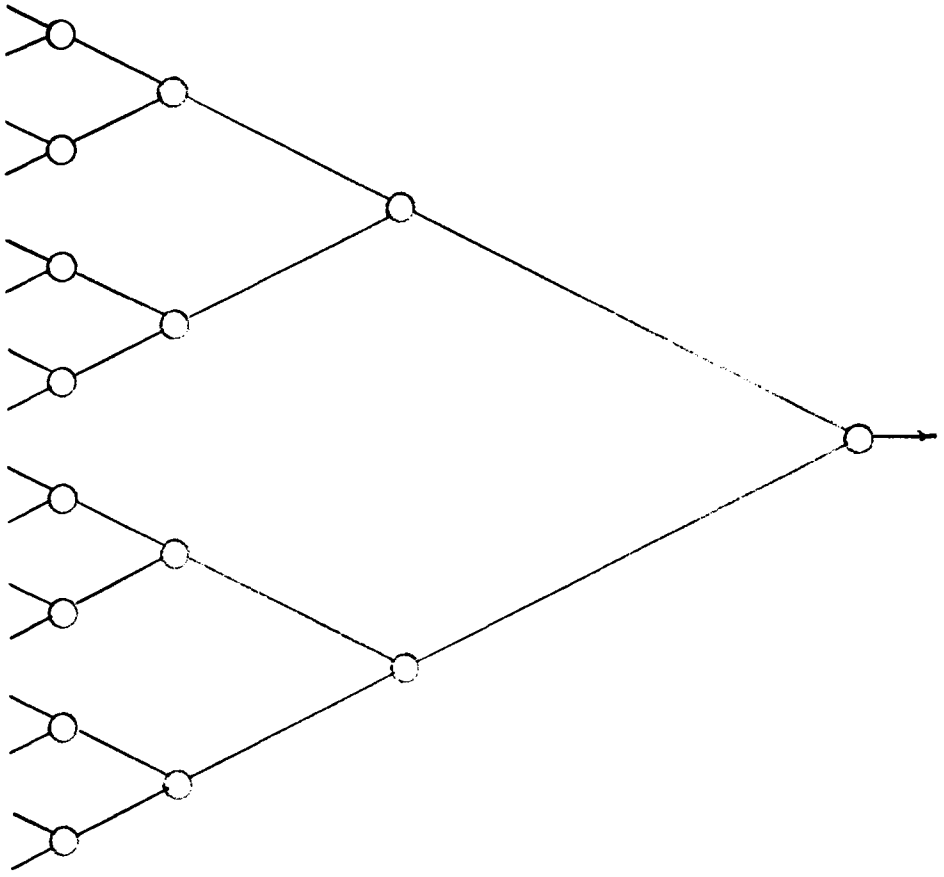


Figure 2. 2-stage Sequential Code Reduction Circuit.

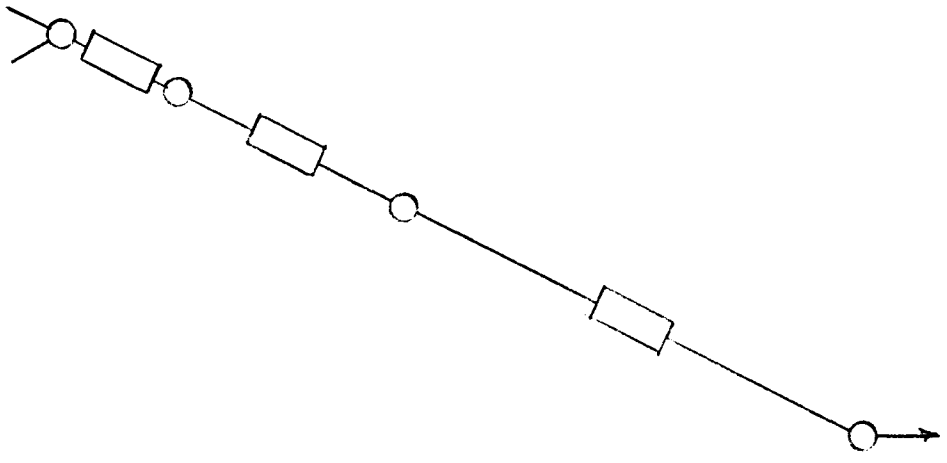
2.3 Sequential code reduction: the basic idea

Essentially what we have done in the circuit of Figure 2 is to use only the upper branch of the combination majority logic decoding tree of the decoder in Figure 1 by inserting a one-word delay between the two levels of majority logic. This concept is illustrated more generally in Figure 3. The trick is to use the cyclic property of the code at each level, not just the last. In general, this allows us to reduce the combinational complexity of the decoder from an exponential function of the number of levels to a linear function, at the cost of a linear increase in buffer storage and decoding time (or delay if additional buffering is used).

The term "sequential code reduction" comes from the fact that we have a different decoding problem for a different code at each stage of the reduction process. Thus, once we obtain s_4, s_4', s_4'', etc. (Figure 2), we have a new decoding problem: single-error-correction for the triple-error-correcting (7,1) code. This is because the new parity checks s_4, s_4', etc., obtained by a nonlinear process, correspond not to words in the dual of the original (7,4) code, but to words in the dual of the (7,1) code. It is important to note that once the new parity checks have been found, we are not tied



(a) Combinational decoding tree



(b) Sequential code reduction

Figure 3. Sequential Code Reduction: Basic Idea

in any way to the decoding method used at the previous stage. In fact, error-trap decoding at the second stage of sequential code reduction of the (7,4) code is probably a better choice than majority logic decoding. And we could have used some other decoding method at the first stage as well. This means that sequential code reduction algorithms take a variety of forms because of the wide choice of decoding schemes at each stage of the decoding process. The choice depends on the subcode being decoded at that stage and on design considerations imposed by the intended application.

SECTION 3

APPLICATION TO FINITE-GEOMETRY CODES

We now consider the application of sequential code reduction to binary cyclic Euclidean-geometry[1,2] and projective-geometry[1,3,4,5,6] codes. In this section, we will only allow sequential code reduction algorithms of the following restricted form: a majority logic decoder with 2t orthogonal parity checks and one majority gate at each stage of the decoding process. The resulting algorithms are probably not optimal with respect to implementation complexity, but they are so simple that it is questionable whether further optimization would be worth the effort. More important for our present purposes, the use of a standard decoding algorithm allows us to make general statements about classes of codes. The class of finite-geometry codes is particularly suitable from this standpoint because a great deal is known about the application of L-step orthogonalization[7,8] to these codes and it is possible -- at least in all cases that we have investigated -- to convert a conventional L-step majority decoding circuit to an L-stage sequential code reduction circuit of the restricted type specified above. The example in the previous section was an instance of such a

conversion. We remark here that L-step orthogonalization does not always decode a finite-geometry code up to its true minimum distance, and this will be true of the corresponding restricted sequential code reduction algorithms. Since one of our purposes here is to compare the relative complexities of L-step orthogonalization and sequential code reduction, we will consider decoding only up to d_{ML}, the minimum distance guaranteed by L-step orthogonalization.

3.1 Euclidean-geometry codes

The points of $EG(m,2^s)$, the m-dimensional Euclidean geometry over $GF(2^s)$, can be taken to be the elements of $GF(2^{ms})$. A k-flat in $EG(m,2^s)$ is the set of points

$$\alpha^j = \alpha^{e_0} + \beta_1\alpha^{e_1} + \ldots + \beta_k\alpha^{e_k},\ \beta_i \in GF(2^s)$$

where $\alpha^{e_1},\ldots,\alpha^{e_k}$ are a fixed set of k elements of $GF(2^{ms})$ that are linearly independent over $GF(2^s)$ and $\beta_1,\ldots,\beta_k$ range over all possible values in $GF(2^s)$. The $(r,s)^{th}$-order binary Euclidean-geometry (EG) code of length $n = 2^{ms} - 1$ is the largest cyclic code whose dual code contains the polynomials corresponding to all r+1-flats in $EG(m,2^s)$ that do not pass through the origin. In what follows, the term "flat" will be used to denote both the point set and the associated polynomial.

Let α be a primitive element of $GF(2^{ms})$ and $h(x)$ the parity check polynomial of the $(r,s)^{th}$-order EG code. Then α^t is a root of $h(x)$ if $W_s(t) \leq r$, where the s-weight of t (denoted $W_s(t)$) is the largest number of multiples of 2^s-1 whose radix 2 expansions are disjoint and covered by the radix 2 expansion of t.[9] (E.g., for s = 2, m = 3, the 2-weight of t = 7 = 000111 is $W_2(7) = 1$ and the 2-weight of t = 15 = 001111 is $W_2(15) = 2$). For s = 1, s-weight reduces to the usual definition of Hamming weight. A combinatorial expression for the degree of $h(x)$, and hence the dimension of the $(r,s)^{th}$-order EG code, has been obtained by Hamada[10] and Lin[11].

The minimum distance guaranteed by L-step orthogonalization for the $(r,s)^{th}$-order EG code of length $n = 2^{ms} - 1$ is

$$d_{ML} = \frac{2^{(m-r)s} - 1}{2^s - 1} .$$

Chen[12] has shown that

$$L = 1 + \left\lceil \log_2 \left(\frac{m}{m-r} \right) \right\rceil$$

steps are sufficient to orthogonalize this code, where $\lceil x \rceil$ denotes the least integer greater than or equal to x, and that this is the minimum number of steps possible. We next consider necessary and sufficient conditions for the conversion of the L-step conventional majority decoding algorithm to a restricted L-stage sequential code reduction algorithm.

The generator of a cyclic code of length n is usually taken to be the code polynomial of least degree that divides all other code polynomials modulo $x^n - 1$. For purposes of the present discussion, it is convenient to generalize this somewhat and say that any code polynomial that divides all code polynomials modulo $x^n - 1$ is a generator of the code. A necessary and sufficient condition for the existence of an L-stage restricted sequential code reduction algorithm is that at each stage of majority decoding there exist a flat that divides each member of a set of 2t flats orthogonal on a flat of lower dimension at the next stage. A sufficient but not necessary condition is that there exist a flat which is a generator of the subcode containing all flats at that stage. Suppose f(x) is a flat in the subcode generated by h(x). Then f(x) is a generator of the subcode if and only if GCD(f(x), $x^n - 1$) = h(x). We will call such a flat a "generator flat".

Suppose at a given stage in the decoding of an $(r,s)^{th}$-order EG code, we are decoding a $(k,s)^{th}$-order EG subcode. A k+1-flat f(x) in the dual of the subcode is a generator of the dual if and only if the roots of f(x) are exactly those α^t for which $W_s(t) \leq k$. It has been verified by computer that there exists a generator k-flat in EG(m,2^s) for all ms $\leq$ 11 and k = 1,...,m. Thus, all $(r,s)^{th}$-order EG codes of length n $\leq$ 2047 can be decoded

by an L-stage sequential code reduction algorithm of the restricted type considered in this section. A comparison of complexity and decoding time for conventional L-step majority decoding and L-stage sequential code reduction for some representative EG codes is given in Table I.

			No. of Maj. Gates		No. of Parity Checks		No. of Buffers		Decoding Time	
r	s	(n,k,t_{ML})	CMD	SCR	CMD	SCR	CMD	SCR	CMD	SCR
1	2	(1023,288,42)	85	2	7,056	168	1	2	T	2T
3	1	(2047,232,127)	255	2	64,516	508	1	2	T	2T
7	1	(4095,3302,15)	241	3	3,375	75	1	3	T	3T
7	1	(8191,5812,31)	3907	3	238,328	186	1	3	T	3T

Table I. Decoding Complexity for Some EG Codes

Although we conjecture that the set of k-flats in $EG(m,2^s)$ contains a generator flat for all k,m and s, we have not been able to show this. However, it will be shown that all cyclic Reed-Muller (RM) codes can be decoded by a restricted sequential code reduction algorithm, although possibly not in the minimum number of stages given by Chen.

It is well known that the r^{th}-order cyclic RM code$^{(7,13,14)}$ (the $(r,1)^{th}$-order EG code) is r+1-step orthogonalizable. In this case, r+1-flats are used to obtain r-flats, r-flats to obtain r-1-flats, etc., until the 0-flat corresponding to $\hat{e}_0$ is obtained. We now prove the following theorem.

Theorem: The r+1-step conventional majority decoding algorithm for the r^{th}-order cyclic RM code can be converted to an r+1-stage restricted sequential code reduction algorithm.

The proof of this theorem requires the following lemmas.

Lemma 1: Let f and g be the k-flats

$$f: \quad \alpha^j = \alpha^{e_0} + \beta_1\alpha^{e_1} + \ldots + \beta_r\alpha^{e_k}, \ \beta_i \varepsilon GF(2)$$

$$g: \quad \alpha^j = \quad\quad \beta_1\alpha^{e_1} + \ldots + \beta_r\alpha^{e_k}, \ \beta_i \varepsilon GF(2)$$

in EG(m,2). Then $f(\alpha^t) = g(\alpha^t)$ for all t such that $W_1(t) = k$.

(Proof) Let h be the k+1-flat

$$h: \quad \alpha^j = \beta_0\alpha^{e_0} + \beta_1\alpha^{e_1} + \ldots + \beta_k\alpha^{e_k}, \ \beta_i \varepsilon GF(2)$$

in EG(m,2). Then $h(\alpha^t) = 0$ for all t such that $W_1(t) \leq k$ and in particular for $W_1(t) = k$. But since β_0 can take only the values 0 and 1, $h(x) = f(x) + g(x)$ and $0 = h(\alpha^t) = f(\alpha^t) + g(\alpha^t)$ from which we see that $f(\alpha^t) = g(\alpha^t)$.

Q.E.D.

Lemma 2: Let f be the k-flat

$$f: \quad \alpha^j = \alpha^{e_0} + \beta_1\alpha^{e_1} + \ldots + \beta_k\alpha^{e_k}, \ \beta_i \varepsilon GF(2)$$

in EG(m,2) where $e_i = e_0+i$ for $i = 1,\ldots,k$. Then $f(\alpha^t) \neq 0$ for all t such that $W_1(t) = k$.

(Proof) First note that since the points $\alpha^{e_0},\ldots,\alpha^{e_k}$ are necessarily linearly independent over GF(2), f does not pass through the origin. By lemma 1, we know that

$f(\alpha^t) = g(\alpha^t)$ if $W_1(t) = k$, where g is the k-flat

$$g: \quad \alpha^j = \beta_1 \alpha^{e_1} + \ldots + \beta_k \alpha^{e_k}, \ \beta_i \varepsilon GF(2).$$

So it suffices to show that $g(\alpha^t) \neq 0$ for all t such that $W_1(t) = k$. If $W_1(t) = k$, the radix 2 expansion of t will be of the form

$$t = 2^{\theta_1} + 2^{\theta_2} + \ldots + 2^{\theta_k}.$$

Then, since the characteristic of the field is two and $\beta_i^{2^{\theta_j}} = \beta_i$, $g(\alpha^t)$ can be written as

$$g(\alpha^t) = \sum_{\beta_i \varepsilon GF(2)} (\beta_1 \alpha^{e_1} + \ldots + \beta_k \alpha^{e_k})^t$$

$$= \sum_{\beta_i \varepsilon GF(2)} \prod_{j=1}^{r} (\beta_1 \alpha^{e_1 2^{\theta_j}} + \ldots + \beta_k \alpha^{e_k 2^{\theta_j}}).$$

Each term in this expression is one of the possible products of k elements from the k x k array

$$
\begin{array}{cccc}
\beta_1\alpha^{e_1 2^{\theta_1}} & \beta_2\alpha^{e_2 2^{\theta_1}} & \cdots & \beta_k\alpha^{e_k 2^{\theta_1}} \\
\beta_1\alpha^{e_1 2^{\theta_2}} & \beta_2\alpha^{e_2 2^{\theta_2}} & \cdots & \beta_k\alpha^{e_k 2^{\theta_2}} \\
\vdots & \vdots & & \vdots \\
\beta_1\alpha^{e_1 2^{\theta_k}} & \beta_2\alpha^{e_2 2^{\theta_k}} & \cdots & \beta_k\alpha^{e_k 2^{\theta_k}}
\end{array}
$$

where one element is chosen from each of the k rows, and the sum is taken over all possible values of $\beta_1,..,\beta_k$. Now note that any product which includes two or more elements from the same column of the array will occur an even number of times in the summation and will therefore not contribute to the sum. This is because if two elements are chosen from the same column, then there is some other column, say the j^{th}, from which no element was selected and two identical terms will result from the summation over β_j.

Clearly the only products that contribute to the sum are those for which one element is selected from each row and column of the array and $\beta_i = 1$ for $i = 1,\ldots,k$. The sum of these terms is just the determinant of the matrix

$$K = \begin{bmatrix} \alpha^{e_1 2^{\theta_1}} & \alpha^{e_2 2^{\theta_1}} & \cdots & \alpha^{e_k 2^{\theta_1}} \\ \alpha^{e_1 2^{\theta_2}} & \alpha^{e_2 2^{\theta_2}} & \cdots & \alpha^{e_k 2^{\theta_2}} \\ \vdots & \vdots & & \vdots \\ \alpha^{e_1 2^{\theta_k}} & \alpha^{e_2 2^{\theta_k}} & & \alpha^{e_k 2^{\theta_k}} \end{bmatrix}$$

Hence $g(\alpha^t) = \det(K)$. Now let $X_j = \alpha^{2^{\theta_j}}$. Then

$$g(\alpha^t) = \det \begin{bmatrix} X_1^{e_1} & X_1^{e_2} & \cdots & X_1^{e_k} \\ X_2^{e_1} & X_2^{e_2} & \cdots & X_2^{e_k} \\ \vdots & \vdots & & \vdots \\ X_k^{e_1} & X_k^{e_2} & & X_k^{e_k} \end{bmatrix}$$

$$= \det \begin{bmatrix} X_1^{e_1} & X_1^{e_1+1} & \cdots & X_1^{e_1+k-1} \\ X_2^{e_1} & X_2^{e_1+1} & \cdots & X_2^{e_1+k-1} \\ \vdots & \vdots & & \vdots \\ X_k^{e_1} & X_k^{e_1+1} & \cdots & X_k^{e_1+k-1} \end{bmatrix}$$

which is a van der Monde determinant. To show that $g(\alpha^t) \neq 0$, it is necessary to verify that $X_i \neq X_j$ for $i \neq j$. But if $X_i = X_j$ then

$$\alpha^{2^{\theta_i}} = \alpha^{2^{\theta_j}} \text{ or } 2^{\theta_j}(2^{\theta_i - \theta_j} - 1) \equiv 0 \pmod{2^m-1}$$

so that $2^{\theta_i - \theta_j} \equiv 1 \pmod{2^m-1}$ which is possible only if $\theta_i = \theta_j$ since $t < 2^m - 1$. Thus $g(\alpha^t) = f(\alpha^t) \neq 0$ for $W_1(t) = k$.

Q.E.D.

(Proof of the theorem) By lemma 2, we can always find a k-flat f in EG(m,2), not passing through the origin, such that $f(\alpha^t) \neq 0$ for all t such that $W_1(t) = k$. We know that it is always possible to find 2t k+1-flats $f_1,\ldots,f_{2t}$ that are orthogonal on f. If f has no roots α^t for which $W_1(t) > k$, then f is a generator flat. So assume that $\alpha^{t_1},\ldots,\alpha^{t_u}$ are all the roots of f(x) for which $W_1(t_i) > k$, $i = 1,\ldots,u$. Since $f_1,\ldots,f_{2t}$ are orthogonal on f, $f_1+c,\ldots,f_{2t}+c$ are orthogonal on f+c for any code word c in the dual of the k^{th}-order RM code. Of course f+c is not in general a flat in the dual of the $(k-1)^{st}$-order RM code, but if it is a generator of the dual then all k-flats can be obtained from f+c, which serves the same purpose. Thus we will have proven the theorem if we can show that it is always possible to find a code word c in the dual of the k^{th}-order RM code such that f+c is a generator of the $(k-1)^{st}$-order code. Every word in the dual of the k^{th}-order RM code

has α^t as a root if $W_1(t) < k+1$. So we choose c to be a word in the dual of the k^{th}-order RM code which also has as roots all α^v such that $W_1(v) > k+1$ and $v \neq t_i$ for $i = 1,\ldots,u$. Then the only roots f and c have in common are the α^t for which $W_1(t) < k$. But since every nonzero element of $GF(2^m)$ is a root of either f or c, this means that the roots of f+c are just those α^t for which $W_1(t) < k$. Hence, f+c is a generator of the dual of the $(k-1)^{st}$-order RM code and this argument holds for each stage of the decoding process from $k = r$ to $k = 0$.

Q.E.D.

This result can be improved using Chen's result on the minimum number of steps required to orthogonalize an $(r,s)^{th}$-order EG code. He showed that whenever $k \leq r$ and $k \leq \frac{m}{2}$, it is always possible to find 2t k-flats orthogonal on a 0-flat. Since all 0-flats are generators, the reduction from the $(k-1)^{st}$-order RM code to $(2^m - 1, 0)$ code (technically the -1^{st}-order RM code) can be accomplished in one stage of sequential code reduction. This leads immediately to the following upper bound on the number of stages of sequential code reduction actually required.

Corollary: Every r^{th}-order RM code can be decoded using a restricted sequential code reduction algorithm of at most $r - v + 2$ stages, where $v = \min(r, \lfloor \frac{m}{2} \rfloor)$ and $\lfloor x \rfloor$ denotes the integer part of x.

We now give an example to illustrate the implementation of restricted sequential code reduction for EG codes.

Example: The 2^{nd}-order RM code of length $2^m - 1 = 31$ is a triple-error correcting (31,16) BCH code that can be orthogonalized in two steps. Using conventional majority logic decoding[15], the decoder requires one buffer, 7 majority gates and 36 parity checks. The corresponding sequential code reduction circuit requires 2 buffers, 2 majority gates and 12 parity checks, and the decoding time is twice that for the conventional decoder.

We know that the dual of the 2^{nd}-order RM code contains all 3-flats that do not pass through the origin, and that we can find $2t = 6$ 3-flat orthogonal on a 2-flat. In particular, the six 3-flats

$$f_1(x) = 1 + x + x^4 + x^8 + x^{12} + x^{13} + x^{15} + x^{17}$$

$$f_2(x) = 1 + x^2 + x^4 + x^9 + x^{12} + x^{15} + x^{22} + x^{27}$$

$$f_3(x) = 1 + x^3 + x^4 + x^{11} + x^{12} + x^{15} + x^{25} + x^{28}$$

$$f_4(x) = 1 + x^4 + x^{12} + x^{14} + x^{15} + x^{18} + x^{20} + x^{30}$$

$$f_5(x) = 1 + x^4 + x^5 + x^6 + x^{12} + x^{12} + x^{15} + x^{16}$$

$$f_6(x) = 1 + x^4 + x^{12} + x^{15} + x^{19} + x^{21} + x^{26} + x^{29}$$

are orthogonal on the 2-flat $f_1'(x) = 1 + x^4 + x^{12} + x^{15}$, and the six 2-flats

$$f_1'(x) = 1 + x^4 + x^{12} + x^{15}$$
$$f_2'(x) = 1 + x^8 + x^{24} + x^{30}$$
$$f_3'(x) = 1 + x^{16} + x^{17} + x^{29}$$
$$f_4'(x) = 1 + x^7 + x^{18} + x^{28}$$
$$f_5'(x) = 1 + x^{20} + x^{21} + x^{22}$$
$$f_6'(x) = 1 + x^9 + x^{11} + x^{13}$$

are orthogonal on the 0-flat $f''(x) = 1$. Since deg $\mathrm{GCD}(f_1'(x), x^{31} - 1) = 6$, $f_1'(x)$ has only six roots from among the 31^{st} roots of unity. But there are exactly six values of t in the range $0 \leq t < 31$ for which $W_1(t) \leq 1$. Then $f_1'(x)$ must be a generator of the dual of the 1^{st}-order RM (31,6) code. Hence the 2-flats $f_1'(x),\ldots,f_6'(x)$ can be expressed as multiples of $f_1'(x)$ modulo $x^{31} - 1$. In particular, $f_i(x) \equiv a_i(x) f_1'(x) \pmod{x^{31} - 1}$ where

$$a_1(x) = 1$$
$$a_2(x) = 1+x^4+x^{12}+x^{15}$$
$$a_3(x) = 1+x+x^2+x^4+x^6+x^{10}+x^{12}+x^{13}+x^{15}+x^{17}+x^{18}+x^{24}$$
$$a_4(x) = 1+x^3+x^4+x^8+x^{16}+x^{19}$$
$$a_5(x) = 1+x+x^2+x^3+x^4+x^5+x^6+x^9+x^{10}+x^{12}+x^{16}+x^{19}+x^{20}+x^{21}+x^{23}+x^{24}$$
$$a_6(x) = 1+x+x^2+x^3+x^5+x^6+x^7+x^8+x^{10}+x^{16}+x^{19}+x^{20}+x^{21}+x^{24}$$

The resulting sequential code reduction circuit is shown in Figure 4.

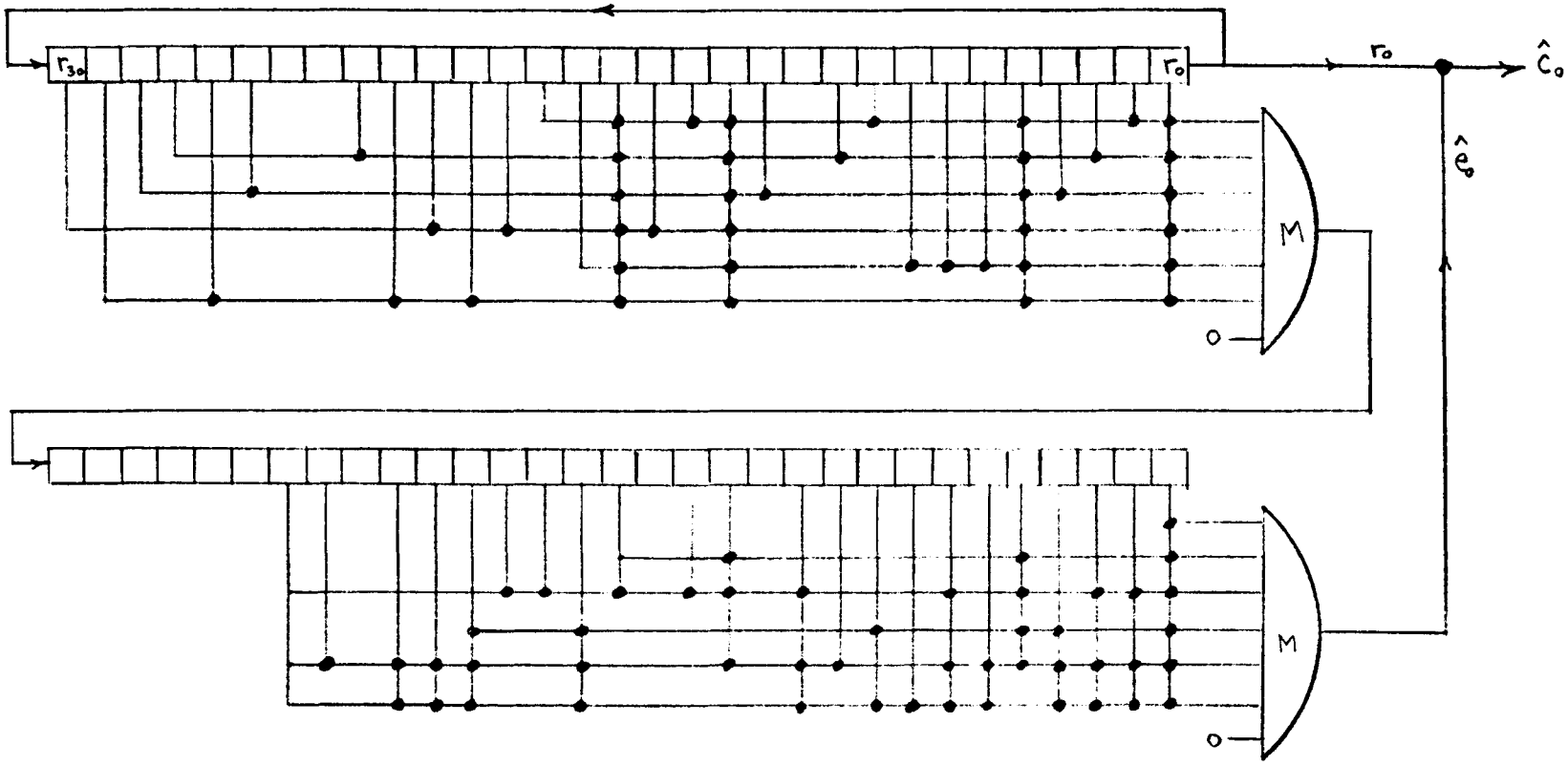


Figure 4. Sequential Code Reduction Circuit for the (31,16) Code.

3.2 Projective-geometry codes

The points of $PG(m,2^s)$, the m-dimensional projective geometry over $GF(2^s)$, can be taken to be the cosets in the multiplicative group of $GF(2^{(m+1)s})$ with respect to the contained multiplicative group of $GF(2^s)$. The coset $\{\alpha^{j_1},\alpha^{j_2},\ldots\alpha^{j_{2^s-1}}\}$ will be denoted by (α^j) where $j = \min(j_1,j_2,\ldots j_{2^s-1})$. Note that under this convention, the cosets are (α^0), $(\alpha^1),\ldots,(\alpha^{n-1})$ where $n = (2^{(m+1)s} - 1)/(2^s - 1)$.

A k-flat in $PG(m,2^s)$ is defined to be the set of all points (cosets)

$$(\alpha^j) = (\beta_0\alpha^{e_0} + \beta_1\alpha^{e_1} +\ldots+ \beta_k\alpha^{e_k}),\ \beta_i\varepsilon GF(2^s)$$

where α^{e_0}, $\alpha^{e_1},\ldots,\alpha^{e_k}$ are a fixed set of k+1 elements of $GF(2^{(m+1)s})$ that are linearly independent over $GF(2^s)$ and $\beta_0,\beta_1,\ldots,\beta_k$ range over all possible values in $GF(2^s)$ except that not all β can be simultaneously zero. With each flat we associate in the natural way a polynomial of degree less than $n = (2^{(m+1)s} -1)/(2^s-1)$. In what follows, the term "flat" will be used to denote both the point set and the associated polynomial. The $(r,s)^{th}$-order projective-geometry (PG) code of length $n = (2^{(m+1)s}-1)/(2^s-1)$ is the largest cyclic code whose dual code contains all r-flats in $PG(m,2^s)$.

Let α be a primitive element of $GF(2^{(m+1)s})$ and h(x) the parity check polynomial of the $(r,s)^{th}$-order PG code. Then $\alpha^{t(2^s-1)}$, $t \neq 0$, is a root of h(x) if $W_s(t(2^s-1)) \leq r$. A combinatorial expression for the degree of h(x), and hence the dimension of the $(r,s)^{th}$-order PG

code, has been obtained by Hamada[10] and Lin[11].

For the $(r,s)^{th}$-order PG code, the minimum distance guaranteed by L-step orthogonalization is

$$d_{ML} = \frac{2^{s(m-r+1)} - 1}{2^s - 1} + 1.$$

Chen[12] has shown that the $(r,s)^{th}$ order PG code of length $n = (2^{(m+1)s} - 1)/(2^s - 1)$ can be orthogonalized in

$$L = 1 + \left\lceil \log_2 \frac{m}{m - r + 1} \right\rceil$$

steps. By an argument analogous to that used for EG codes, it can be shown that a sufficient condition for the conversion of the conventional L-step majority decoding algorithm to the corresponding restricted sequential code reduction algorithm is the existence of generator flats in $PG(m,2^s)$. It has been verified by computer that there exists a k-flat that is a generator flat in $PG(m,2^s)$ for all $(m + 1)s \leq 12$ and $k = 1,\ldots,m$. Thus, all PG codes of length $n < 5461$ can be decoded by a restricted L-stage sequential code reduction algorithm. A comparison of complexity and decoding time for conventional L-step majority decoding and L-stage sequential code reduction is given in Table II for some representative PG codes.

r	s	(n,k,t_{ML})	No. of Maj. Gates		No. of Parity Checks		No. of Buffers		Decoding Time	
			CMD	SCR	CMD	SCR	CMD	SCR	CMD	SCR
2	2	(85,68,2)	5	2	16	8	1	2	T	2T
2	2	(341,195,10)	21	2	400	40	1	2	T	2T
2	2	(1365,483,42)	85	2	7,056	168	1	2	T	2T
2	3	(4681,3105,36)	73	2	5,184	144	1	2	T	2T

Table II. Decoding Complexity for Some PG Codes.

SECTION 4

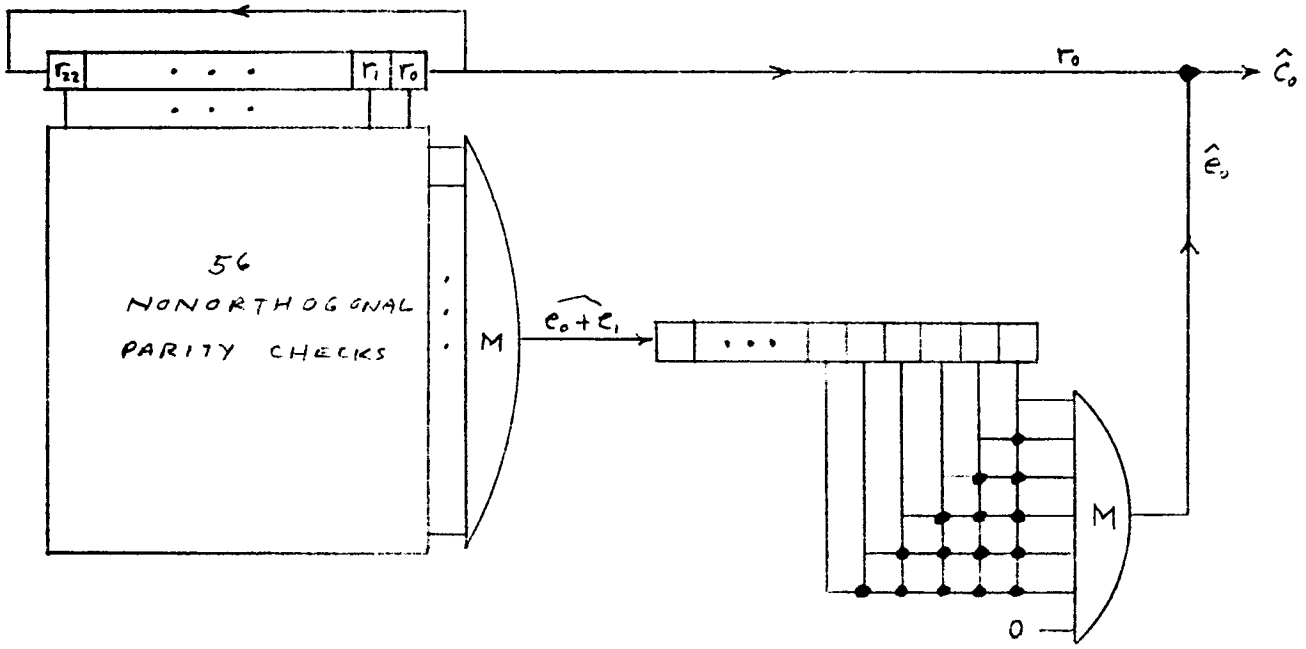
TWO DECODING OPTIONS FOR THE GOLAY CODE

In this section, we present two implementation options for sequential code reduction of the (23,12) Golay triple-error-correcting code. This code is not L-step orthogonalizable, so a sequential code reduction algorithm of the restricted type considered in the previous section is not applicable. Since the Golay code has only two cyclic subcodes, the (23,1) and (23,0) codes, the decoder will have at most two stages.

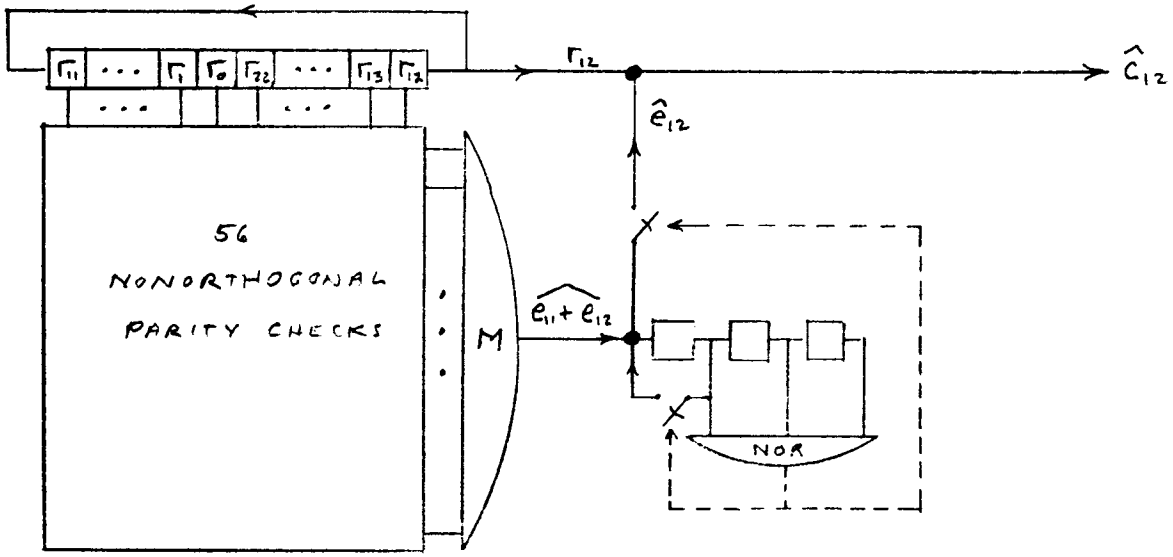
The decoder shown in Figure 5(a) consists of a majority decoder with 56 nonorthogonal parity checks and one majority gate at the first stage, and a majority decoder with 6 orthogonal parity checks and one majority gate at the second stage.

The decoder shown in Figure 5(b) has the same first stage as the decoder in Figure 5(a), but utilizes an error-trap decoder at the second stage. This configuration, a slight variation of a decoder devised by Schmandt[16], decodes in time T rather than 2T. This is possible because the decoder need only decode the

k = 12 information bits and because the 11-error-correcting (23,1) code is only required to correct triple errors. (This configuration requires that the check bits of a code word be transmitted before the information bits.)



(a) Majority:: majority decoder



(b) Majority:: error-trap decoder

Figure 5. Two Sequential Code Reduction Circuits for the Golay Code

SECTION 5

DISCUSSION

We have shown that sequential code reduction is applicable to a number of cyclic codes and that significant reductions in decoder complexity can be obtained at the cost of a modest increase in decoding time. In particular, we have shown that all finite-geometry codes of length $n \leq 2047$ can be decoded by a restricted L-stage sequential code reduction algorithm using 2t orthogonal parity checks and one majority gate at each stage. We conjecture that any L-step orthogonalizable code can be decoded up to d_{ML} by a restricted algorithm of this form. If our conjecture proves out, then it becomes possible to consider the use of very long L-step orthogonalizable codes, on the order of 10^4 or even 10^5 bits in length, in practical error-control systems. Very long codes are important in practice because channels with memory require long codes and most real channels are of this type. It might be argued that very long L-step orthogonalizable codes are inferior to, say, very long BCH codes, but from a practical point of view this is irrelevant because it is not economically feasible to decode very long BCH codes. The only long codes used in practice are those constructed from short codes by interleaving, concatenation, etc.

These codes are also inferior to efficient cyclic codes of the same length and the overall decoding process is often complex. Sequential code reduction mav make very long L-step orthogonalizable codes, and perhaps other cyclic codes as well, competitive with coding systems based on interleaving and concatenation. We might note here that majority sequential code reduction looks particularly promising because of its inherent capability to correct many error patterns of weight greater than t. The longer the code, the more significant this extra capacity becomes.

ADDENDUM

While this paper was in preparation, some related work by researchers in the U.S.S.R.[17,18] was brought to our attention[19]. To our knowledge, their results have not yet been published in English.

REFERENCES

1. Rudolph, L. D., "Geometric Configurations and Majority Logic Decodable Codes," M.E.E. Thesis, University of Oklahoma, Norman (1964).

2. Weldon, E. J. Jr., "Euclidean Geometry Cyclic Codes," Proceedings of the Conference on Combinational Mathematics and its Applications (Bose and Dowling, Eds.), The University of North Carolina Press, Chapel Hill, N.C. (1967).

3. Rudolph, L. D., "A Class of Majority Logic Decodable Codes," IEEE Trans. Info. Theory, IT-13, 305-307 (1967).

4. Goethals, J. M. and P. Delsarte, "On a Class of Majority-Logic Decodable Cyclic Codes," IEEE Trans. Info. Theory, IT-14, 182-188 (1968).

5. Weldon, E. J. Jr., "Difference Set Cyclic Codes," Bell System Tech. J., 7, 1045-1055 (1966).

6. Weldon, E. J., Jr., "New Generalizations of the Reed-Muller Codes -- Part II: Nonprimitive Codes," IEEE Trans. Info. Theory, IT-14, 199-205, (1968).

7. Reed, I. S., "A Class of Multiple-Error-Correcting Codes and the Decoding Scheme," IEEE Trans. Info. Theory, IT-4, 38-49 (1954).

8. Massey, J. L., "Threshold Decoding", M.I.T. Press, Cambridge, Mass. (1963)

9. Kasami, T., S. Lin and W. W. Peterson, "Some Results on Cyclic Code which are Invariant under the Affine Group," Scientific Report, AFCRL-66-622, Air Force Cambridge Research Laboratory, Bedford, Mass. (1966).

10. Hamada, N., "The Rank of the Incidence Matrix of Points of d-flats in Finite Geometries," J. Sci., Hiroshima University, Vol. 32, 381-396 (1968)

11. Lin, S., "On the Number of Information Symbols of Polynomial Codes," to appear.

12. Chen, C. L., "Note on Majority-Logic Decoding of Finite Geometry Codes", to appear.

13. Muller, D. E., "Application of Boolean Algebra to Switching Circuit Design and to Error Detection," IEEE Trans. Electron. Computers, EC-3, 6-12 (1954).

14. Kasami, T., S. Lin and W. W. Peterson, "New Generalizations of the Reed-Muller Codes, Part I: Primitive Codes," IEEE Trans. Info. Theory, IT-14, 189-198 (1968).

15. Berlekamp, E., "Algebraic Coding Theory", McGraw-Hill (1968).

16. Schmandt, F., private communication (1971).

17. Yu. M. Shtar'kov, "A Scheme for Majority Decoding with L-step Orthogonalization" Prob. Per Inform (translation).

18. Kolesnik, V. D. and Ye, T. Mironchikov, "Decoding of Cyclic Codes", Moscow, Svyaz' (1968). (title translated).

19. Schmandt, F., private communication (1972).