A Simplified Procedure for Decoding the (23,12) and (24,12) Golay Codes

T. K. Truong and J. K. Holmes
Communications Systems Research Section

I. S. Reed and X. Yin
University of Southern California, Department of Electrical Engineering

very large scale integration (VLSI)

In this article, a simplified procedure is developed to decode the three possible errors in a (23,12) Golay codeword. A computer simulation shows that this algorithm is modular, regular, and naturally suitable for both ~~VLSI~~ and software implementation. An extension of this new decoding procedure is used also to decode the 1/2-rate (24,12) Golay code, thereby correcting three and detecting four errors.

I. Introduction

The Golay code is a very useful code particularly for applications in which a parity bit is added to each codeword to yield a rate 1/2 code. The (24,12) Golay code is currently supported by the DSN. The 24-bit Golay code is also attractive for use on the DSN uplink where the spacecraft uses software decoding with the on-board computer. This code has been used on a number of communication links in the past, including the Voyager imaging system link.

This decoding problem originated with an investigation of the Digital Communication Terminal (DCT) communication link performance using the (23,12) Golay code. Coding analysts at JPL evaluated the properties of the Golay code, achieving a solution to the relatively simple decoding of the (23,12) and (24,12) Golay codes.

In the present article the BCH decoding algorithm described in [5] is extended to correct all three possible (correctable) errors of the code. This new decoding procedure is based on the fact that if one of the three errors in the block code of 23 digits can be canceled first, then the BCH decoding algorithm can be used to correct the remaining two errors.

This new Golay decoder is quite simple and similar to the Weldon decoding procedure [2]. One of the advantages of this algorithm over previous methods (see [1–3]) is that the new decoding algorithm is easily understood and readily implemented.

II. The (23,12) Binary Golay Code

It is not difficult to show that $\mathbf{g}(x) = x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1$ is an irreducible polynomial over $GF(2)$. Thus, there exists an element $\alpha \in GF(2^{11})$ such that $\mathbf{g}(\alpha) = 0$. Hence, the elements of $GF(2^{11})$ are found in the following set:

$$GF(2^{11}) = \left\{ a_0 + a_1 \alpha + a_2 \alpha^2 + \ldots + a_{10} \alpha^{10} \mid a_0, a_1, \ldots, a_{10} \in GF(2) \right\} \qquad (1)$$

Note also that α is a primitive twenty-third root of unity in $GF(2^{11})$. This fact shows that $\mathbf{g}(x)$ generates a cyclic BCH block code of length 23, called the Golay code.

The (23,12) Golay code is a perfect or close-packed code in the sense that the codewords and their 3-error correction spheres exhaust the vector space of 23-bit binary vectors. It is shown in [5] that the (23,12) Golay code is, besides being a cyclic BCH code, a quadratic-residue code. Since the minimum distance of the code is $d = 7$, one has the inequality $2t + 1 \leqslant d$, where t is the number of errors to be corrected. Hence the (23,12) Golay code allows for the correction of $t \leqslant 3$ errors.

The codewords of a Golay code over $GF(2)$ are expressed first as the coefficients of a polynomial. In such a representation a codeword is represented by

$$C(x) = \sum_{i=0}^{22} c_i x^i \tag{2}$$

where $c_i \in GF(2)$ and x is an indeterminant.

The generator polynomial of a Golay code as discussed above is an irreducible polynomial and given by

$$\mathbf{g}(x) = \prod_{i=0}^{10} (x - \alpha^{2^i})$$

$$= x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1 \tag{3}$$

Now let polynomials

$$I(x) = c_{22}x^{22} + c_{21}x^{21} + \ldots + c_{11}x^{11} \tag{4}$$

and

$$P(x) = c_{10}x^{10} + c_9x^9 + \ldots + c_1x + c_0 \tag{5}$$

be the information and the parity-check polynomials of a codeword $C(x)$. Then the codeword in Eq. (2) is represented by

$$C(x) = I(x) + P(x) \tag{6}$$

To be a (23,12) cyclic BCH Golay codeword, $C(x)$ must also be a multiple of the generating polynomial $\mathbf{g}(x)$. That is,

$$C(x) = q(x)\mathbf{g}(x) \tag{7}$$

Polynomial $P(x)$ in Eq. (6) is obtained by dividing $I(x)$ by $\mathbf{g}(x)$, i.e.,

$$I(x) = q(x)\mathbf{g}(x) + \mathbf{r}(x) \tag{8}$$

where $\mathbf{r}(x)$ is a remainder polynomial of degree less than 11. Then one sets $P(x) = \mathbf{r}(x)$. Thus by Eqs. (6), (7), and (8) the following identities are true:

$$q(x)\mathbf{g}(x) = I(x) + \mathbf{r}(x) = I(x) + P(x) = C(x) \tag{9}$$

A code generated in this manner is a cyclic BCH code with parity check polynomial $P(x) = \mathbf{r}(x)$.

III. The Decoder for a (23,12) Golay Code

A simple BCH decoding algorithm is developed in [5] to decode a (23,12) Golay code with only two or less errors. To illustrate this method, define

$$E(x) = e_{22}x^{22} + e_{21}x^{21} + \ldots + e_1x + e_0 \tag{10}$$

to be the error polynomial. Then the received codeword has the form

$$R(x) = C(x) + E(x) \tag{11}$$

Suppose that e errors occur in the received codeword $R(x)$ and assume that $2t \leqslant d - 1$. The decoder begins by dividing the received codeword $R(x)$ by the generator polynomial $\mathbf{g}(x)$. That is,

$$R(x) = V(x)\mathbf{g}(x) + S(x) \tag{12}$$

where $deg[S(x)] < deg[\mathbf{g}(x)]$ with $deg[\cdot]$ denoting degree of polynomial. Also by Eqs. (7) and (11),

$$R(x) = q(x)\mathbf{g}(x) + E(x) \tag{13}$$

Hence, by Eqs. (12) and (13) the syndrome polynomial $S(x)$ is found to be

$$S(x) = M(x)\mathbf{g}(x) + E(x) \tag{14}$$

where $M(x) = q(x) + V(x)$. Since α and α^3 are both roots of $\mathbf{g}(x)$, one has

$$\begin{aligned} s_1 &\triangleq E(\alpha) = S(\alpha) \\ s_3 &\triangleq E(\alpha^3) = S(\alpha^3) \end{aligned} \tag{15}$$

where s_1 and s_3 are called the syndromes of the code.

The error-locator polynomial is defined by

$$\sigma(z) = \prod_{\beta = \textit{error-locations}} (1 - \beta z) \tag{16}$$

For two errors, $\sigma(z)$ is given by the polynomial [5]

$$\sigma(z) = 1 + s_1 z + \left(s_1^2 + \frac{s_3}{s_1}\right) z^2 \tag{17}$$

where s_1 and s_3 are the syndromes defined in Eq. (15).

After reception, the 2-error correcting decoder computes by Eq. (15) the syndromes s_1 and s_3, as well as the error-locator polynomial given in Eq. (17). Depending on the number of errors, this BCH decoding algorithm (described in [5]) satisfies the following scheme:

$$\sigma(z) = \begin{cases} 1 & \text{if } s_1 = s_3 = 0, \\ & \text{then no error} \\ 1 + s_1 z & \text{if } s_3 = s_1^3, \\ & \text{then one error} \\ 1 + s_1 z + \left(s_1^2 + \dfrac{s_3}{s_1}\right) z^2 & \text{if } s_1 \neq 0 \text{ and} \\ & s_3 \neq 0, \\ & \text{then two errors} \end{cases} \tag{18}$$

Note that the roots of $\sigma(z)$ are the inverse locations of the $t = 2$ errors.

From Eq. (18), it is evident that if there are no more than two errors, the errors can be located by the roots of $\sigma(z)$. Suppose $\sigma(z)$ does not have both its roots in the multiplicative subgroup of the field $GF(2^{11})$ consisting of the 23 roots of unity, namely, $\mathbf{G} = \{\alpha^i \mid 0 \leqslant i \leqslant 22\}$. Then this decoding procedure fails. Since a Golay code is perfect or close-packed, this implies that the above BCH decoding scheme in Eq. (18) detects more than two errors, namely, three errors.

IV. Decoder for Correcting Three Errors

For simplicity, let the transmitted error and received code be re-expressed, respectively, by the binary vectors: $\underline{c} = (c_0, c_1, c_2, \ldots, c_{22})$, $\underline{e} = (e_0, e_1, e_2, \ldots, e_{22})$ and $\underline{r} = (r_0, r_1, r_2, \ldots, r_{22})$.

Definition 1. Let the Hamming norm or weight of a binary vector $\underline{x} = (x_1, x_2, \ldots, x_n)$ be designated by $\|\underline{x}\|$. Then the set

$$\mathbf{T}_i \triangleq \{\underline{e} \mid \|\underline{e}\| = i\} \tag{19}$$

is the set of error vectors of weight i.

Definition 2. In terms of the Hamming norm or the weight, the Hamming distance between two vectors $\underline{x}$ and $\underline{y}$ is defined by

$$d(\mathrm{x,y}) \triangleq \|\mathrm{x} - \mathrm{y}\| \tag{20}$$

The above concepts are now used to prove the following theorem:

Theorem. Let $\underline{e}_4$ be any error vector of weight 4 and $\underline{c}$ be any code vector of the (23,12) Golay code. Then

$$\underline{x} \triangleq \underline{c} + \underline{e}_4 = \underline{c}_1 + \underline{e}_3 \tag{21}$$

where $\underline{c}_1$ is some other code vector and $\underline{e}_3$ is some error vector of weight 3. In other words, adding an error vector of Hamming weight 4 to a codeword of a Golay code produces a 23-bit vector which is equal to some other codeword plus an error vector of weight 3.

Proof: First, it is well known that the (23,12) Golay code is close-packed. That is, $\mathbf{B}_{23} = \mathbf{C}_0 \cup \mathbf{C}_1 \cup \mathbf{C}_2 \cup \mathbf{C}_3$ where $\cup$ denotes set union, $\mathbf{C}_0$ is the set of Golay code words, $\mathbf{B}_{23}$ is the set of all binary vectors of length 23 and

$$\mathbf{C}_i = \{\underline{x} \mid \underline{x} = \underline{c} + \underline{e}, \underline{c} \in \mathbf{C}_0, \underline{e} \in \mathbf{T}_i\} \qquad \text{for } i = 1, 2, 3$$

Hence, for any $\underline{c} \in \mathbf{C}_0$ and $\underline{e}_4 \in \mathbf{B}_{23}$, one has $\underline{x} = \underline{c} + \underline{e}_4 \in \mathbf{B}_{23}$, and $\underline{x} \notin \mathbf{C}_0$. Thus, by the close-packed nature of the code there exists $\underline{c}_1 \in \mathbf{C}_0$ and an error vector $\mathbf{e}$ such that

$$\underline{x} = \underline{c}_1 + \underline{e} \tag{22}$$

with

$$\|\underline{e}\| = \|\underline{x} + \underline{c}_1\| \leqslant 3 \tag{23}$$

Secondly, it is known that in a Golay code any nonzero codeword has a minimum weight of 7. Hence, by hypothesis and Eq. (22), one has the equalities

$$\|\underline{x} + \underline{c}_1\| = \|\underline{c}_1 + \underline{e}_4 + \underline{c}\| = \|\underline{e}\| \tag{24}$$

Also, since

$$\|\underline{x}+\underline{y}\| \geqslant \Big| \|\underline{x}\| - \|\underline{y}\| \Big| \qquad \text{and} \qquad \min_{\underline{c}_1 \neq \underline{c}} \|\underline{c}_1 + \underline{c}\| = 7$$

one has by Eq. (24) the inequalities

$$\|\underline{x} + \underline{c}_1\| = \|\underline{e}\| \geqslant \Big| \|\underline{c}_1 + \underline{c}\| - \|\underline{e}_4\| \Big| \geqslant \Big|7 - 4\Big| = 3 \quad (25)$$

Thus by combining Eqs. (23) and (25), $\|\underline{e}\| = \|\underline{e}_3\| = 3$, and the theorem is proved. A geometric view of this proof is shown in Fig. 1.

Remark: The above theorem and its proof generalize to any close-packed error correcting code. However, since there is only one other nontrivial multiple error correcting perfect code, the (11,6) Golay code over *GF*(3), such generality is somewhat academic.

Suppose the codeword $\underline{c} = (c_0, c_1, \ldots, c_{22})$ of the (23,12) Golay code is transmitted, and that the error vector $\underline{e} = (e_0, e_1, \ldots, e_{22})$ occurs with weight $t \leqslant 3$, where $d = 7 \geqslant 2t + 1$. Also, let the received vector be

$$\underline{r} = \underline{c} + \underline{e} = (r_0, r_1, \ldots, r_{n-1}) \quad (26)$$

In this terminology, the new decoding method can be described as a recursive algorithm, as described below.

If $\underline{r}$ is corrupted by an error pattern $\underline{e}$ of weight $t \leqslant 2$, i.e., $\|\underline{e}\| \leqslant 2$, the BCH decoding method of Eq. (18) can correct all patterns of two or fewer errors. On the other hand, if the transmitted code $\underline{c}$ is corrupted by an error pattern $\underline{e}$ of weight three, i.e., $\|\underline{e}\| = 3$, then the scheme in Eq. (18) can be extended to correct three errors.

The first step in the decoding procedure is to cancel one error from $\underline{e}$. The second step is to correct the remaining two errors using the BCH method. To describe this algorithm, let $\underline{u}_1 = (1, 0, 0, \ldots, 0)$ be the "unit" 23-tuple vector. $\underline{u}_1$ has only one nonzero component, located at the first position. The sum of $\underline{r}$ and $\underline{u}_1$ is

$$\underline{r} + \underline{u}_1 = \underline{r}_1 = \underline{c} + \underline{e} + \underline{u}_1 \quad (27)$$

If one of the three errors in $\underline{e}$ is located at the first coordinate position, then $\|\underline{e} + \underline{u}_1\| = 2$, and the BCH decoding method for two errors in Eq. (18) can be used to correct the remaining two errors. However, if the first component of $\underline{e}$ is zero, then, by Theorem 1, $\underline{c} + \underline{e} + \underline{u}_1 = \underline{c}_1 + \underline{e}_1$, where $\|\underline{e}_1\| = 3$ and $\underline{c}_1 \in C$ such that $\underline{c}_1 \neq \underline{c}$. In this case the BCH decoding method in Eq. (18) again can be used to detect the presence of three errors in received code vector $\underline{r}$.

Let ρ be the permutation which shifts the contents of the register right by one bit. Hence, a shift of $\underline{u}_1$ right by one bit is given by $\rho(\underline{u}_1) = \underline{u}_2 = (0, 1, 0, \ldots, 0)$. The sum of $\underline{r}_1$, $\underline{u}_1$, and $\underline{u}_2$ is given by

$$\underline{r}_2 = \underline{r}_1 + \underline{u}_1 + \underline{u}_2 = \underline{c} + \underline{e} + \underline{u}_2 \quad (28)$$

The same decoding procedure that was used on $\underline{r}_1$ is applied now to vector $\underline{r}_2$.

The procedure used above for the first and second coordinates of $\underline{r}$ is repeated recursively for at most 12 steps. For each step either one can correct all three errors or detect the fact that three errors still remain. Thus, if one shifts $\underline{u}_1$ through all of the information bits and one error is not canceled by the twelfth shift of $\underline{u}_1$, namely

$$\rho^{12}(\underline{u}_1) = (0,0,0,0,0,0,0,0,0,0,0,0,1,0,\ldots,0)$$

it is evident that all three errors exist only in the parity bits and therefore the information bits are not in error and can be decoded directly.

The overall decoding of a (23,12) Golay code is summarized by the following steps:

1. Apply the BCH procedure of Eq. (18) to decode the received 23-bit code vector $\underline{r}$. If an error vector $\underline{e}$ occurs with weight $t \leqslant 2$, i.e., $\|\underline{e}\| \leqslant 2$, the error pattern can be corrected. If $\|\underline{e}\| = 3$, the BCH procedure in Eq. (18) fails, but detects that three errors must corrupt vector $\underline{r}$.
2. Next, the first received information bit is inverted. Then the BCH procedure is applied again to the received code $\underline{r}$ now modified in the first bit. If $\|\underline{e}\| \leqslant 2$, the above inversion of the first bit corrects this bit. Thus the BCH method can now be used to correct the other two errors. On the other hand, if the first information bit was originally correct, the BCH method detects the fact that the three errors still remain in the codeword.
3. Repeat step 2 by inverting the second, third, ..., twelfth bits. If the BCH method still detects three errors with the received vector changed at the twelfth position, then all errors are confined to the parity check section.

Example: The following example illustrates how the decoding algorithm corrects 3 errors in one received codeword.

Encoding

Let $I(x)$ and $g(x)$ be the information and generator polynomials. Then $P(x)$, the parity polynomial, is found by $P(x) = I(x) \bmod g(x)$. That is,

$$I(x) = x^{22} + x^{20} + x^{18} + x^{17} + x^{15} + x^{14} + x^{12} + x^{11}$$

$$g(x) = x^{11} + x^{9} + x^{7} + x^{6} + x^{5} + x + 1$$

$$P(x) = x^{10} + x^{9} + x^{7} + x^{6} + x^{5} + x^{4} + x^{3} + 1$$

Hence the codeword vector is

$$\underline{c} = 1\,0\,1\,0\,1\,1\,0\,1\,1\,0\,1\,1 \quad 1\,1\,0\,1\,1\,1\,1\,1\,0\,0\,1$$

Channel-Noise

The three random errors are introduced at the 20th, 18th, and 9th bit of the codeword. Then the error pattern is given by

$$\underline{e} = 0\,0\,1\,0\,1\,0\,0\,0\,0\,0\,0\,0 \quad 0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0$$

and the received codeword is

$$\underline{r} = 1\,0\,0\,0\,0\,1\,0\,1\,1\,0\,1\,1 \quad 1\,0\,0\,1\,1\,1\,1\,1\,0\,0\,1$$

Decoding

Loop 1. First the syndrome $S(x)$ is found from codeword $R(x)$ by $R(x) \bmod g(x)$. That is,

$$R(x) = x^{22} + x^{17} + x^{15} + x^{14} + x^{12} + x^{11} + x^{10} + x^{7}$$
$$+ x^{6} + x^{5} + x^{4} + x^{3} + 1$$

$$S(x) = x^{9} + x^{7} + x^{6} + x^{5} + x^{4} + x^{3} + 1$$

$$s_1 = S(\alpha) = \alpha^{9} + \alpha^{7} + \alpha^{6} + \alpha^{5} + \alpha^{4} + \alpha^{3} + 1$$

$$s_3 = S(\alpha^3) = \alpha^{9} + \alpha^{8} + \alpha^{7} + \alpha^{6} + \alpha^{3} + \alpha^{2} + \alpha^{1}$$

$$s_1^3 = [S(\alpha)]^3 = \alpha^{9} + \alpha^{8} + \alpha^{7} + \alpha^{5} + \alpha^{1}$$

Since $s_1 \neq s_3 \neq 0$, $s_1^3 \neq s_3$, there are at least two errors. Thus,

$$\sigma(z) = 1 + s_1 z + \left(s_1^2 + \frac{s_3}{s_1}\right) z^2$$

Multiplying both sides by s_1 one has

$$\sigma'(z) = s_1 + s_1^2 z + \left(s_1^3 + s_3\right) z^2$$

Using the Chien search on $\sigma'(z)$, two roots cannot be found. Hence there are three errors.

Loop 2. Next invert the first bit of the received codeword. Use the same procedure as in Loop 1. Two roots still cannot be found.

Loop 3. Same as Loop 2, but invert the second bit.

Loop 4. After inverting the third bit, the syndrome polynomial and syndromes are given by

$$S(x) = x^{10} + x^{6} + x^{5} + x^{3} + x^{2}$$

$$s_1 = S(\alpha) = \alpha^{10} + \alpha^{6} + \alpha^{5} + \alpha^{3} + \alpha^{2}$$

$$s_3 = S(\alpha^3) = \alpha^{8} + \alpha^{4}$$

Since $s_1^3 = \alpha^{10} + \alpha^{8} + \alpha^{3} + \alpha^{2} + \alpha^{1}$, then $s_1 \neq s_3 \neq 0$ and $s_1^3 \neq s_3$. Thus one has for the error-locator polynomial

$$\sigma'(z) = (\alpha^{10} + \alpha^{6} + \alpha^{5} + \alpha^{3} + \alpha^{2})$$
$$+ (\alpha^{10} + \alpha^{9} + \alpha^{8} + \alpha^{1} + \alpha^{0}) z$$
$$+ (\alpha^{10} + \alpha^{4} + \alpha^{3} + \alpha^{2} + \alpha^{1}) z^2$$

From the Chien search, it is found that

$$\sigma'(\alpha^5) = (\alpha^{10} + \alpha^{6} + \alpha^{5} + \alpha^{3} + \alpha^{2})$$
$$+ (\alpha^{10} + \alpha^{9} + \alpha^{8} + \alpha^{1} + \alpha^{0}) \alpha^{5}$$
$$+ (\alpha^{10} + \alpha^{4} + \alpha^{3} + \alpha^{1} + \alpha^{0}) \alpha^{10} = 0$$

$$\sigma'(\alpha^{14}) = (\alpha^{10} + \alpha^{6} + \alpha^{5} + \alpha^{3} + \alpha^{2})$$
$$+ (\alpha^{10} + \alpha^{9} + \alpha^{8} + \alpha^{1} + \alpha^{0}) \alpha^{14}$$
$$+ (\alpha^{10} + \alpha^{4} + \alpha^{3} + \alpha^{1} + \alpha^{0}) \alpha^{28} = 0$$

Hence, one has two roots, namely, $\beta_1 = \alpha^5$ and $\beta_2 = \alpha^{14}$. Thus all errors are located. Summarizing, one has

$$\text{first error position} = \alpha^{21}$$

$$\text{second error position} = \beta_1^{-1} = \alpha^{23-5} = \alpha^{18}$$

$$\text{third error position} = \beta_2^{-1} = \alpha^{23-14} = \alpha^{9}$$

After correctng the received codeword according to these error locations, the successfully decoded codeword is given by

$$\hat{\underline{c}} = 10101101101111011111001$$

The flowchart of this new algorithm is shown in Fig. 2. In a computer simulation, several hundred random codes with three or less errors were created and decoded perfectly with an average speed of 1.67 seconds per code. These results are shown in more detail in Table 1. The above algorithm is extended to the 1/2-rate (24,12) Golay code in the next section.

V. The (24,12) Golay Code

A (24,12) Golay codeword can be formed by adding an even or odd parity-check bit to the (23,12) Golay codeword. It is shown easily that such a (24,12) Golay code has the minimum distance $d_{\min} = 8$. Thus, the new decoding algorithm for the extended (24,12) Golay code can be used to correct three or less errors and to detect the presence of four errors.

There is no loss in generality to assume that the parity of a (24,12) Golay codeword is even. That is, the sum of the 24 bits modulo 2 is equal to zero. Assume during transmission that four errors are added to the codeword. There are two cases to consider, as follows:

1. If the four errors occur in the first 23 bits, then by the theorem in Section IV, the addition of an error vector of Hamming weight 4 to a codeword produces a 23-bit vector which is equal to some other (23,12) Golay codeword plus an error vector of weight 3. Thus if the new decoding algorithm in Section IV is applied to the first 23 bits, an error vector of weight 3 is added to the received codeword. As a consequence, the parity of the (24,12) codeword also is changed. Hence, by checking the parity of the decoded codeword, the decoder detects the presence of four errors.
2. On the other hand, if three errors occur in the first 23 bits, and one error occurs in the parity bit, the new decoding algorithm corrects the three errors in the first 23 bits. The parity of the 23-bit decoded codeword now differs from the received parity bit. Hence, the decoder detects the presence of four errors.

The extended decoding algorithm for the (24,12) Golay code is summarized as follows: apply the simplified decoding algorithm to the first 23 bits. If the number of errors is less than 3, the decoding procedure terminates normally. If the number of errors is greater than or equal to 3, the parity of the decoded codeword is compared with the received parity bit. If they are different, the decoder detects four errors.

The detailed flowchart of the above decoding procedure is shown in Fig. 3. Finally, a comparison of the average computer times to decode the (23,12) and (24,12) Golay codes is given in Table 1. The decoding speeds for the (24,12) Golay codes are slightly lower due to the possibility of the parity bit being in error.

VI. Conclusion

An extended BCH algorithm is obtained for correcting three errors in a (23,12) Golay code. This procedure is based on the fact that if one bit is reversed in a codeword which has three errors, this codeword changes to another codeword which still has three errors. Hence, if one of the three errors can be canceled first, then the standard BCH decoding procedure can be used to correct the remaining two errors. A computer simulation shows that this procedure is very modular and naturally suitable for both software and VLSI implementation.

It is shown in the flowchart of Fig. 3 that the above new algorithm can be extended to decode the 1/2-rate (24,12) Golay code. The decoding algorithm for the (24,12) Golay code corrects 3 or less errors and detects the presence of 4 errors.

Acknowledgment

The authors wish to thank Mr. Rodney Pau at the University of Southern California for his help in computer programming.

References

[1] T. Kasami, "A Decoding Procedure for Multiple-Error-Correcting Cyclic Codes," *IEEE Trans. Information Theory*, IT-10, pp. 134–139, April 1964.

[2] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*, New Jersey: Prentice-Hall, Inc., 1983.

[3] F. J. MacWilliams, "Permutation Decoding of Systematic Codes," *Bell Syst. Tech. J.*, vol. 43, part 1, pp. 485–505, January 1964.

[4] F. J. MacWilliams and W. J. A. Sloane, *The Theory of Error-Correcting Codes*, Amsterdam: North-Holland, 1977.

[5] E. R. Berlekamp, *Algebraic Coding Theory*, New York: McGraw-Hill, 1968.

Table 1. The computer time needed for decoding

Number of errors	Average computer CPU time, sec
0	0.001
1	0.173
2	0.232
3	1.67

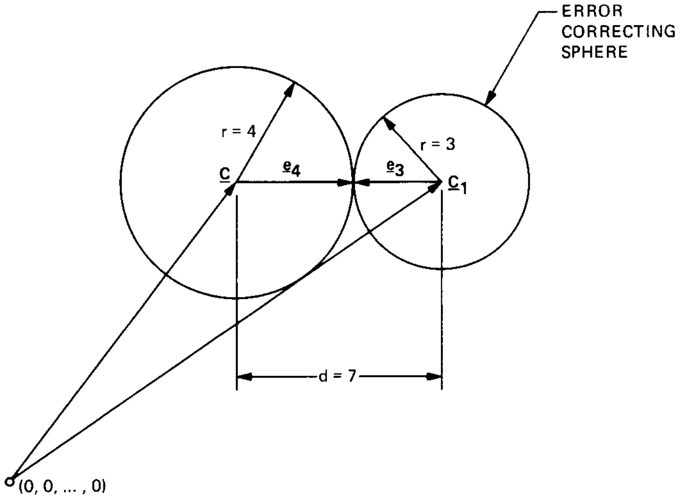

Fig. 1. A geometric view of the proof of Theorem 1.

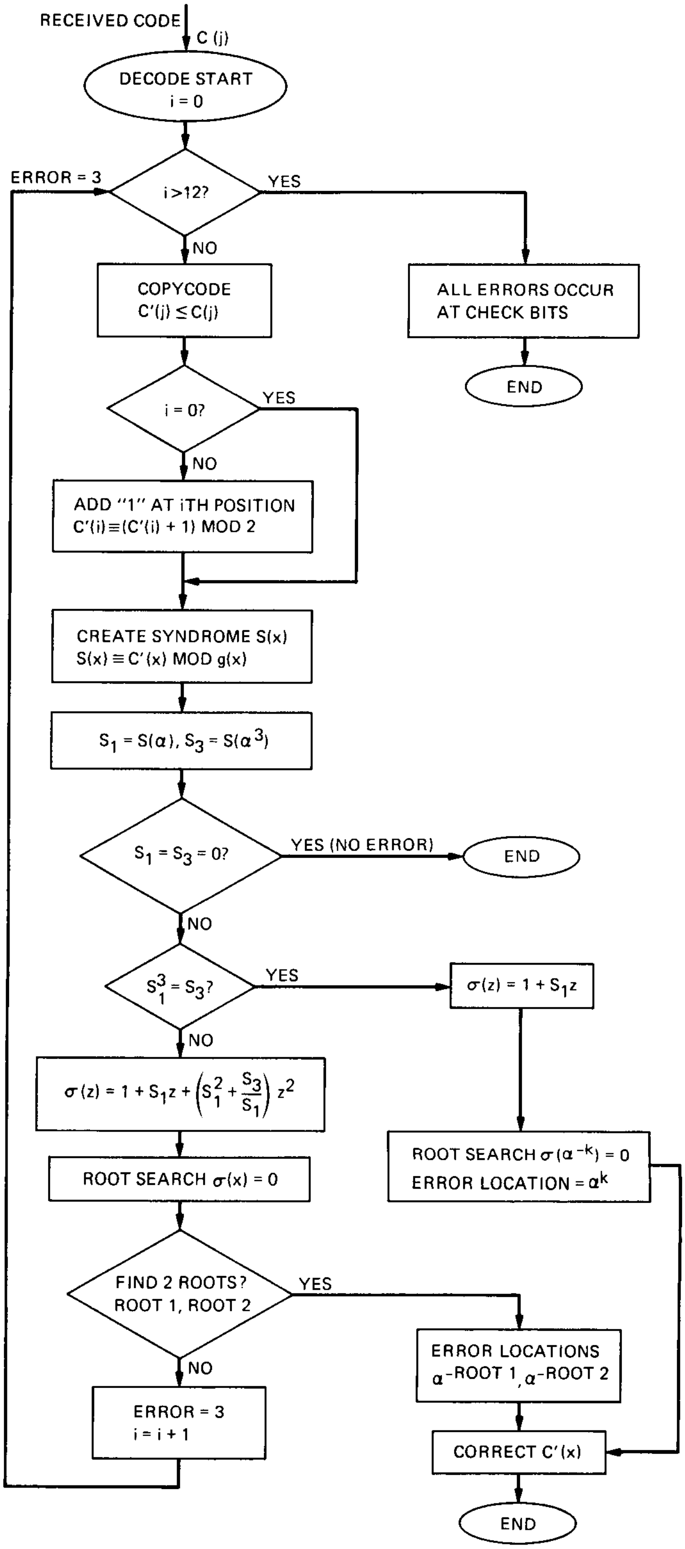

Fig. 2. Flowchart of the decoding algorithm.

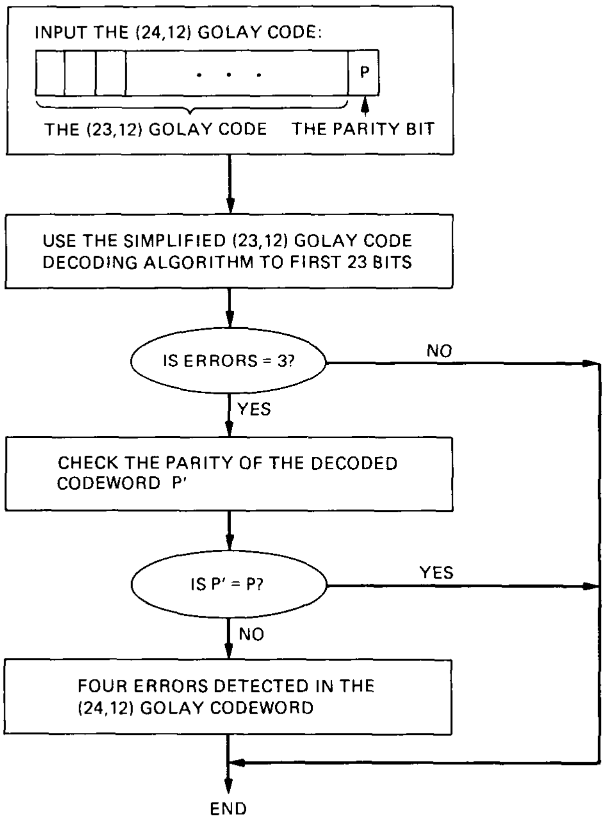
INPUT THE (24,12) GOLAY CODE:
P
THE (23,12) GOLAY CODE
THE PARITY BIT
USE THE SIMPLIFIED (23,12) GOLAY CODE DECODING ALGORITHM TO FIRST 23 BITS
IS ERRORS = 3?
NO
YES
CHECK THE PARITY OF THE DECODED CODEWORD P'
IS P' = P?
YES
NO
FOUR ERRORS DETECTED IN THE (24,12) GOLAY CODEWORD
END

Fig. 3. The decoding flowchart of the (24,12) Golay code.