the (positive) bent function $f^{+}(x) = \mathrm{tr}(ax^{2l-1})$, with $a \in G^{+}$ and $K(a) = -1$;
the (positive) difference set $\Omega^{+} = \{x \in \mathrm{GF}(2^{2l}) | f^{+}(x) = 0\} = \{0, \theta_1, \cdots, \theta_{r-1}\}$
with $r = 2^{2l-1} + 2^{l-1}$, which is a union of $\{0\}$ and $2^{l-1}+1$ classes mod G^{+};
the code $D^{+} = \{(\mathrm{tr}(u\theta_1), \cdots, \mathrm{tr}(u\theta_{r-1})) | u \in \mathrm{GF}(2^{2l})\}$, which is a two-weight projective code with $(2^l - 1)(2^{l-1}+1)$ words of weight 2^{2l-2} and $2^{2l-1} - 2^{l-1}$ words of weight $2^{2l-2} + 2^{l-1}$;
the corresponding strongly regular graph, which is a positive latin square graph.

B

The cyclic multiplicative subgroup G^{-} of order $2^l + 1$ generated by β in $\mathrm{GF}(2^{2l})$;
the (negative) bent function $f^{-}(x) = \mathrm{tr}(bx^{2l+1})$, with $b \in G^{-}$ and $b \neq 1$;
the (negative) difference set $\Omega^{-} = \{x \in \mathrm{GF}(2^{2l}) | f^{-}(x) = 0\} = \{0, \delta_1, \cdots, \delta_{s-1}\}$
with $s = 2^{2l-1} - 2^{l-1}$, which is a union of $\{0\}$ and $2^{l-1} - 1$ classes mod G^{-};
the code
$D^{-} = \{(\mathrm{tr}(u\delta_1), \cdots, \mathrm{tr}(u\delta_{s-1})) | u \in \mathrm{GF}(2^{2l})\}$, which is a two-weight projective code with $(2^l + 1)(2^{l-1} - 1)$ words of weight 2^{2l-2} and $2^{2l-1} + 2^{l-1}$ words of weight $2^{2l-2} - 2^{l-1}$;
the corresponding strongly regular graph, which is a negative Latin-square graph.

References

[1] L. D. Baumert and R. J. McEliece, "Weights of irreducible cyclic codes," *Inform. Contr.*, vol. 20, pp. 158–175, 1972.

[2] E. R. Berlekamp and O. Moreno, "Extended double-error correcting binary Goppa codes are cyclic," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 817–818, 1973.

[3] J. H. Conway and N. J. A. Sloane, "Sphere packings, lattices and groups," in *Grundlehren der Mathematischen Wissenschaften*. New York: Springer-Verlag, 1988, p. 290.

[4] J. F. Dillon, "Elementary Hadamard difference sets," in *Proc. Sixth SECCGTC*, F. Hoffman *et al.* (Eds.). Winnipeg: Utilitas Math., 1975.

[5] ____, "Elementary Hadamard difference sets," Ph.D. thesis, Univ. Maryland, College Park, MD, 1974.

[6] A. Dür, "The weight distribution of double-error-correcting Goppa codes," in *Proc. A.A.E.C.C4* (Karlsruhe) 1986.

[7] C. F. Gauss, *Disquisitiones Arithmeticae*. Leipzig: Fleischer, 1801; English translation, Yale University Press, 1966.

[8] V. D. Goppa, "Codes and information," *Uspekhi Mat. Nauk*, vol. 39, pp. 77–120; *Russian Math. Surveys*, vol. 39, pp. 87–141, 1984.

[9] R. Hartshoren, *Algebraic Geometry*. New York: Springer, 1977.

[10] T. Honda, "Isogeny classes of abelian varieties over finite fields," *J. Math. Soc. Japan*, vol. 20, pp. 83–95, 1968.

[11] G. Lachaud and J. Wolfmann, "Sommes de Kloosterman, courbes elliptiques et codes cycliques en caractéristique 2," *C. R. Acad. Sci. Paris (I)*, vol. 305, pp. 881–883, 1987.

[12] R. Lidl and H. Niederreiter, "Finite fields," in *Encyclopedia of Mathematics and Its Applications*. Reading, MA: Addison Wesley, 1983, vol. 20.

[13] R. J. McEliece, "Correlation properties of sets of sequences derived from irreducible cyclic codes," *Inform. Contr.*, vol. 45, pp. 18–25, 1980.

[14] ____, *Finite Fields for Computer Scientists and Engineers*. Boston, MA: Kluwer, 1986.

[15] F. J. McWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam: North-Holland, 1977.

[16] F. J. McWilliams and J. Seery, "The weight distribution of some minimal cyclic codes," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 796–806, 1981.

[17] J. C. C. M. Remijn and H. J. Tiersma, "A duality theorem for the weight distribution of some cyclic codes," *IEEE Trans. Inform. Theory*, vol. IT-34, no. 4, pt. II, pp. 1348–1351, Sept. 1988.

[18] R. Schoof, "Nonsingular plane cubic curves over finite fields," *J. Combinatorial Theory*, vol. 46, pp. 183–211, 1987.

[19] J. H. Silverman, *The Arithmetic of Elliptic Curves*. New York: Springer, 1986.

[20] J. Tate and P. Deligne, *Modular Functions of One Variable*. Berlin: Springer, 1975, pp. 53–73.

[21] K. K. Tzeng and K. Zimmermann, "On extending Goppa codes to cyclic codes," *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 712–716, 1975.

[22] W. C. Waterhouse, "Abelian varieties over finite fields," *Ann. Scient. Ec. Norm. Sup.*, vol. 4, no. 2, pp. 521–560, 1969.

[23] J. Wolfmann, "Codes projectifs a deux ou trois poids associes aux hyperquadriques d'une geometrie finie," *Discrete Math.*, vol. 13, pp. 185–211, 1975.

[24] ____, "Codes projectifs à deux poids caps complets et ensembles de difference," *J. Comb. Theory*, vol. A23, pp. 208–222, 1977.

On High-Speed Decoding of the (23, 12, 7) Golay Code

SHYUE-WIN WEI AND CHE-HO WEI, SENIOR MEMBER, IEEE

Abstract—**An algebraic decoding method for triple-error-correcting binary BCH codes applicable to complete decoding of the (23,12,7) Golay code has been proved recently. A modified step-by-step complete decoding algorithm of this Golay code is introduced, which needs less shift-operations than Kasami's error-trapping decoder. Based on the algorithm, a high speed hardware decoder of this code is proposed.**

I. Introduction

The (23, 12, 7) Golay code is the only known triple-error-correcting binary perfect code, thus a complete decoding algorithm can be easily achieved if any combination of three or fewer errors can be corrected. Since this Golay code is known as a cyclic code, the algebraic method for decoding cyclic codes, such as Kasami's error-trapping decoder and systematic-search decoder [1], can be applied to decode this Golay code. Another algebraic method, known as the step-by-step decoding algorithm, was proposed by Massey in 1965 for BCH codes [2]. This method involves changing the received symbols one at a time with testing to determine whether the weight of the error pattern has been reduced. By the step-by-step decoding method for the (23, 12, 7) Golay code first proposed in 1966 [3], this method cannot tell the difference between the cases of two errors and three errors. Thus, this method has two disadvantages: a) the need to consider the temporary correction of two errors, b) requirement of a large amount of shift-operations to complete the correction process. Recently, Elia [4] proved that the algebraic method for decoding triple-error-correcting binary BCH codes is also applicable to complete decoding of the (23, 12, 7) Golay code. Using some results of [4], we show that a modified fast step-by-step algebraic decoding algorithm can also be employed for the cyclic Golay codes.

II. Complete Decoding Algorithm

Consider the (23, 12, 7) Golay code with generator polynomial $g(x) = x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1$; the encoded codeword $c(x)$ can be obtained by $g(x)K(x)$ if $K(x)$ is a polynomial associated with 12 information-bits. Moreover, the codeword $c(x)$ can also be generated in systematic form by repermuting the generator matrix or by $K(x)x^{11} + t(x)$, where $t(x)$ indicates

Manuscript received August 3, 1988; revised February 27, 1989. This work was supported by the National Science Council of the Republic of China through research project NSC. 77-0404-E009-08.

The authors are with the Institute of Electronics and Center for Telecommunications Research, National Chiao Tung University, Hsin Chu, Taiwan, Republic of China.

IEEE Log Number 8933619.

0018-9448/90/0500-0692$01.00 ©1990 IEEE

the remainder polynomial of $K(x)x^{11}$ divided by $g(x)$. The roots of this generator polynomial are confirmed as $(\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^6, \alpha^8, \alpha^9, \alpha^{12}, \alpha^{13}, \alpha^{16}, \alpha^{18})$, where α is the primitive 23rd root of unity [5]. The received polynomial is expressed as $r(x) = r_0 + r_1 x + \cdots + r_{22}x^{22}$. By using three roots α, α^3, and α^9, we can compute three syndrome values [4]:

$$\begin{aligned} S_1 &= r(\alpha), \\ S_3 &= r(\alpha^3), \\ S_9 &= r(\alpha^9). \end{aligned} \tag{1}$$

Clearly, $T_3 = T_9 = 0$, if the number of errors is less than two, where $T_3 = (S_1)^3 + S_3$ and $T_9 = (S_1)^9 + S_9$. Property 4' of [2] shows that $T_3 \neq 0$ if the weight of error pattern is 2 or 3, and $S_1 \neq 0$ if the weight of error pattern is 1 or 2. Moreover, as confirmed by computer simulation, we observe that T_9 and S_1 are not equal to zero when the number of errors is three. Elia [4] also proved that $S_3 + S_1[(T_3)^2 + T_9/T_3]^{1/3} = 0$ if two errors have occurred. Therefore, if a new variable $F \in GF(2^{11})$ is defined as

$$F = \begin{cases} S_3 + S_1\left[(T_3)^2 + T_9/T_3\right]^{1/3}, & \text{for } T_3 \neq 0 \\ 0, & \text{for } T_3 = 0 \end{cases} \tag{2}$$

then, the different weights of error patterns can be distinguished from one another in terms of the relations among S_1, T_3 (or T_9) and F. That is,

- if there is no error, then $S_1 = T_3 = T_9 = F = 0$;
- if there is one error, then $S_1 \neq 0$, $T_3 = T_9 = 0$ and $F = 0$;
- if there are two errors, then $S_1 \neq 0$, $T_3 \neq 0$, $T_9 \neq 0$ and $F = 0$;
- if there are three errors, then $S_1 \neq 0$, $T_3 \neq 0$, $T_9 \neq 0$ and $F \neq 0$.

To abridge the decoding algorithm, according to these results, three decision-bits can also be defined in the following:

$$h_1^0 = 1, \qquad \text{if } S_1 = 0, \tag{3}$$

$$h_2^0 = 1, \qquad \text{if } T_3 = 0, \left(\text{or}, h_2^0 = 1 \text{ if } T_9 = 0\right), \tag{4}$$

$$h_3^0 = 1, \qquad \text{if } F = 0. \tag{5}$$

Furthermore, these decision-bits can be included in a decision vector as

$$H^0 = \left(h_1^0, h_2^0, h_3^0\right). \tag{6}$$

Then, the number of errors can be correctly determined in terms of the pattern of vector H^0 if and only if the weight of the received error pattern is three or less. That is, $H^0 = (1,1,1)$ indicates that no errors have occurred; $H^0 = (0,1,1)$ indicates that one error has occurred; $H^0 = (0,0,1)$ indicates that two errors have occurred; and $H^0 = (0,0,0)$ indicates that three errors have occurred. Since the Golay code is a cyclic code, the received word can be cyclically shifted without changing the relationship among its syndrome values. That is, the decision vector H^0 is changed only by changing the weights of the error patterns. Also, it has been shown that if the first position of $r(x)$ can be decoded correctly for all correctable error patterns, then the entire word can be decoded correctly with the same circuitry [6]. Thus, we define

$$S_i^1 = S_i + 1 \qquad i = 1, 3, 9, \tag{7}$$

where S_i^1 are the syndrome values of $r(x) + 1$. From S_i^1, we can obtain

$$T_3^1 = \left(S_1^1\right)^3 + S_3^1 \tag{8}$$

$$T_9^1 = \left(S_1^1\right)^9 + S_9^1 \tag{9}$$

$$F^1 = \begin{cases} S_3^1 + S_1^1\left[\left(T_3^1\right)^2 + T_9^1/T_3^1\right]^{1/3}, & \text{for } T_3^1 \neq 0 \\ 0, & \text{for } T_3^1 = 0. \end{cases} \tag{10}$$

Similarly, some decision-bits can be given in the following:

$$h_1^1 = 1 \text{ if } S_1^1 = 0 \tag{11}$$

$$h_2^1 = 1 \text{ if } T_3^1 = 0 \left(\text{or}, h_2^1 = 1 \text{ if } T_9^1 = 0\right) \tag{12}$$

$$h_3^1 = 1 \text{ if } F^1 = 0 \tag{13}$$

and,

$$H^1 = \left(h_1^1, h_2^1, h_3^1\right). \tag{14}$$

Theorem 1: If the weight of a received error pattern of the (23,12,7) Golay code is three or less, then the error pattern can be corrected by a step-by-step decoding algorithm.

Proof:

Case 1) If the weight of the received error pattern is 1, then $H^0 = (0,1,1)$. Consider temporarily changing the received information digits $r_{22}, r_{21}, \cdots, r_0$ one at a time. Suppose r_j is an erroneous bit, then changing r_j will reduce the weight of the error pattern and hence $H^1 = (1,1,1)$. Conversely, suppose r_j is correct, then changing r_j will increase the weight of the error pattern to two and hence, $H^1 = (0,0,1)$.

Case 2) If the weight of the received error pattern is 2, then $H^0 = (0,0,1)$. Consider temporarily changing the received information digits $r_{22}, r_{21}, \cdots, r_1$ one at a time. Suppose r_j is the first met erroneous bit, then changing r_j will reduce the weight of the error pattern and hence, $H^1 = (0,1,1)$. Conversely, suppose r_j is correct, then changing r_j will increase the weight of the error pattern to 3 and hence, $H^1 = (0,0,0)$. As the first met erroneous bit is found and corrected, then this case becomes Case 1.

Case 3) If the weight of the received error pattern is 3, then $H^0 = (0,0,0)$. Consider temporarily changing the received information digits $r_{22}, r_{21}, \cdots, r_2$ one at a time. Suppose r_j is the first met erroneous bit, then changing r_j will reduce the weight of the error pattern and hence, $H^1 = (0,0,1)$. Conversely, suppose r_j is correct, then changing r_j will increase the weight of the error pattern to 4. If $c(x)$ is the sent word and $e(x)$ the error pattern, then $r(x) + x^j = c(x) + e(x) + x^j$, and the weight of $e(x) + x^j$ is 4. Due to the Golay code G being a 3-perfect one, there is a unique word $c_1(x)$ in G such that $r(x) + x^j = c_1(x) + e_1(x)$ where the weight of $e_1(x)$ is at most 3 and the information on that weight is given by H^1. It follows that the word $c_2(x) = e(x) + e_1(x) + x^j$ belongs to G because it is equal to $c(x) + c_1(x)$. Considering the weightsof $c_2(x)$, $e(x) + x^j$ and $e_1(x)$ we deduce that the only possible weight for $e_1(x)$ is 3 because the minimum weight of G is 7. So we get $H^1 = (0,0,0)$. (The fact is also given by [7].) As the first met erroneous bit is found and corrected, then this case becomes Case 2. □

In summary, a received word can be correctly decoded if the weight of its received error pattern is three or less.

Using the previous theorem, the complete decoding algorithm of the (23,12,7) Golay code is described as follows.

1) Calculate initial syndrome values S_i $(i = 1,3,9)$ and then obtain H^0.

2) If $H^0 = (1,1,1)$, read out $r(x)$ and end this algorithm; otherwise, go to next step.
3) Shift $r(x)$, and calculate S_i^1 $(i = 1,3,9)$ and H^1.
4) If $H^0 = (0,1,1)$ and $H^1 = (1,1,1)$, then change the magnitude of the first position of shifted $r(x)$, refresh S_i and H^0 by setting $S_i = S_i^1$ and $H^0 = H^1$, and go to Step 2); otherwise, go to next step.
5) If $H^0 = (0,0,1)$ and $H^1 = (0,1,1)$, then change the magnitude of the first position of shifted $r(x)$, refresh S_i and H^0 by setting $S_i = S_i^1$ and $H^0 = H^1$, and go to Step 7); otherwise, go to next step.
6) If $H^0 = (0,0,0)$ and $H^1 = (0,0,1)$, then change the magnitude of the first position of shifted $r(x)$, refresh S_i and H^0 by setting $S_i = S_i^1$ and $H^0 = H^1$. Go to next step.
7) If all the 12 information digits have been decoded, then the decoding of $r(x)$ is completed; otherwise, go to Step 3).

The modified step-by-step algorithm needs only 35 shift-operations for decoding one received word, where 23 shift-operations are used for calculating the initial syndrome values S_i in Step 1) and the other 12 shift-operations are used for correcting the errors in information part.

III. Hardware Decoder

This modified step-by-step decoding algorithm can be easily implemented by hardware circuits. Fig. 1 shows the functional block diagram of the hardware decoder. It is partitioned into three parts: syndrome calculation circuit, comparison circuit and decision circuit. The decoder is similar to a new decoder of binary BCH codes presented recently [8]. The syndrome calculation circuit is used to calculate the syndrome values S_i^l $(l = 0,1;$ $i = 1,3,9)$, and the comparison circuit is used to determine the decision bits h_j^l $(l = 0,1;$ $j = 1,2,3)$. The decision circuit is used to perform the operations in Steps 4–6. It can be realized by a logical circuit or a read-only-memory (ROM) circuit of size $2^6 \times 1$ bits. Table I shows the truth table of the decision circuit. According to the decision bits, the decision circuit can tell whether the decoding bit of $r(x)$ is erroneous or not. If the corresponding bit is judged to be an erroneous bit, the decoder sends a correcting-bit E_c to change its magnitude, and then refresh the syndrome values and decision bits. The syndrome calculation circuit and decision circuit are quite simple. The design of the comparison circuit is described as follows.

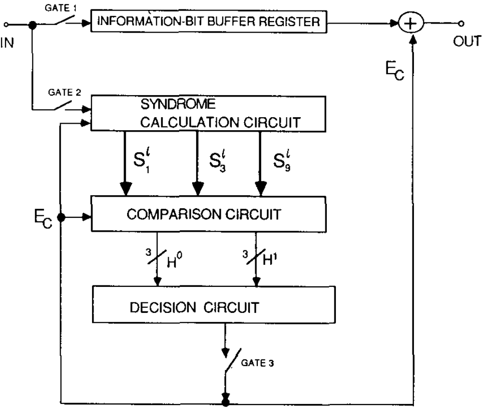

Fig. 1. Functional block diagram of decoder. → is 11-bit bus line, $\not\to$ is 3-bit bus line, → is 1-bit signal line.

TABLE I
Truth Table of the Decision Circuit

Input						Output	Comment
h_1^0	h_2^0	h_3^0	h_1^1	h_2^1	h_3^1	E_c	
0	1	1	1	1	1	1	Find the first error
0	0	1	0	1	1	1	Find the second error
0	0	0	0	0	1	1	Find the third error
Other cases						0	

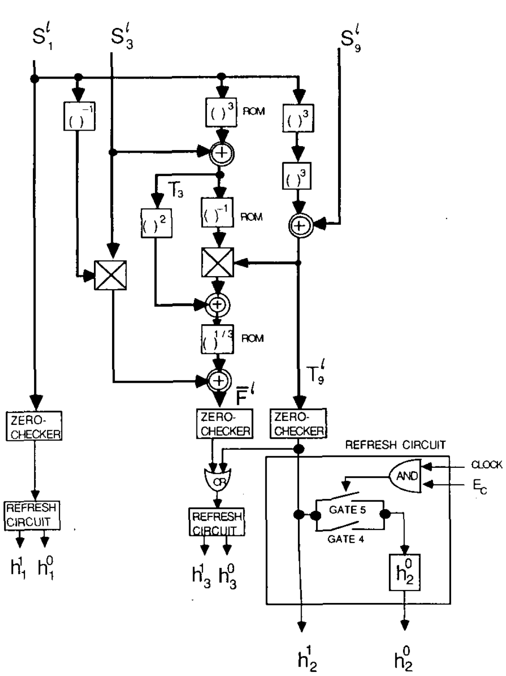

Fig. 2. Comparison circuit of (23, 12, 7) Golay code. → is 11-bit bus line, → is 1-bit signal line, ⊠ is cellular-array multiplier, ⊕ is adder in $GF(2^{11})$.

To determine these decision bits, the values of T_3^l, T_9^l, and F^l $(l = 0,1)$ in $GF(2^{11})$ must be computed first. The add operation in $GF(2^{11})$ is quite simple which can be accomplished by using a set of 11 pieces of 2-input Exclusive-OR gates. To perform the multiplication operation in $GF(2^{11})$, a static cellular-array multiplier can be employed [9], where the computation time of a multiplication for cellular-array multiplier needs 22 gate delays. Moreover, the calculation of power of an element in $GF(2^{11})$ (e.g., $(\alpha^j)^3$) can be implemented by using ROM circuits of $2^{11} \times 11$ bits with table-lookup. The ROM circuits can reduce the computation time and may be easily implemented in VLSI circuits. At the present time, the access time of a ROM can be reduced to under 35 ns (e.g., XL46HC64 SpeedPROM). Similarly, functions $(\cdot)^{1/3}$ and $(\cdot)^{-1}$ can also be implemented by ROM circuits since the cubic root and the inverse of any element in $GF(2^{11})$ are also in $GF(2^{11})$. Furthermore, to shorten the computation path, we define $\bar{F}^l = F^l / S_1^l$ (since $S_1^l \neq 0$ when errors have occurred, $\bar{F}^l = 0$ implies $F^l = 0$). Then, the block diagram of the comparison circuit can be implemented as Fig. 2. Finally, after finding the values of S_1^l, T_9^l and $\bar{F}^l$, the decision bits h_j^l can be determined by using three simple zero-checkers, one of which consists of one 11-input NOR gate. In Fig. 2, the decision bits h_j^0 must be refreshed if an erroneous bit is found. Therefore, three refresh circuits are cascaded with the zero-checkers. In the refresh circuits, gate 4 is used to save the initial decision bits in the storage stage while gate 5 is used to do the refreshing operation.

The proper operation sequence of the decoder is similar to the new decoder of [8]. Since the comparison and decision circuits are static logic circuits, only 35 clock cycles are required for decoding a received word. The decoding speed of this decoder is determined by the period of the clock, which is dominantly determined by the propagation time (i.e., calculation time) of the comparison circuit. In the longest computation path in Fig. 2, it needs to access ROM circuits three times and to perform three additions and one multiplication. Assume the delay time of a multiplication for a cellular-array multiplier is 100 ns and the access time for a ROM circuit is 50 ns. Then, the calculation time of the comparison circuit can be accomplished within 300 ns and thus, a clock rate at 3 MHz can be achieved. This means that the decoder can work at a rate up to $(23/35)\times 3 \cong 2$ Mb/s.

IV. Conclusion

Kasami's error-trapping decoder is one of the best known decoders for decoding the (23,12,7) Golay code. In Kasami's decoder 46 shift-operations are required for decoding one completed received word, while the new step-by-step method only requires 35 shift-operations if the code word is in systematic form. Thus, this new step-by-step decoder is faster than the Kasami's decoder with the same clock rate. Consequently, this new decoding algorithm is suitable for hardware implementation, and a high speed decoder of this code is presented. This decoder requires also 35 clock cycles for decoding one word, and can work at a data rate up to 2 Mb/s. If the Chien's search method is employed, it needs two circuits to calculate the syndrome values and the coefficients of the error location polynomial. The complexities of these circuits are comparable to the syndrome calculation circuit and the comparison circuit of the new step-by-step decoder. However, the Chien's search method still needs a complex circuit to search or solve for the roots of the error location polynomial [5, 10]. Therefore, the new step-by-step decoder is less complex than the Chien's-search method implemented by hardware circuits.

Acknowledgment

The authors would like to thank the reviewers for their valuable comments and suggestions.

References

[1] S. Lin and D. J. Costello, Jr., *Error Control Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[2] J. L. Massey, "Step-by-step decoding of the Bose-Chaudhuri-Hocquenghem codes," *IEEE Trans. Inform. Theory*, vol. IT-11, pp. 580-585, 1965.

[3] R. T. Chien and V. Lum, "On Golay's perfect codes and step-by-step decoding," *IEEE Trans. Inform. Theory*, vol. IT-12, pp. 403-404, 1966.

[4] M. Elia, "Algebraic decoding of the (23,12,7) Golay code," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 150-151, 1987.

[5] E. R. Berlekamp, *Algebraic Coding Theory*. Laguna Hills, CA: Aegean Park Press, 1984.

[6] W. W. Peterson and E. J. Weldon, Jr., *Error-Correcting Codes*, 2nd ed. Cambridge, MA.: MIT Press, 1972.

[7] I. S. Reed, T. K. Truong, X. Y. Yin, and J. K. Holmes, "A simplified procedure for decoding (23,12) and (24,12) Golay codes," TDA Progress Report 42-97, Jan.-Mar. 1989, Jet Propulsion Laboratory, Pasadena, CA.

[8] S. W. Wei and C. H. Wei, "High speed hardware decoder for double-error-correcting binary BCH codes," *IEE Proc.*, vol. 136, pt. I., no. 3, 1989, pp. 227-231.

[9] B. A. Laws, Jr. and C. K. Rushforth, "A cellular-array multiplier for GF(2^m)," *IEEE Trans. Comput.*, vol. C-20, pp. 1573-1578, 1971.

[10] C. L. Chen, "Formulas for the solutions of quadratic equations over GF(2^m)," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 792-794, 1982.

There is No Binary [25,8,10] Code

ØYVIND YTREHUS AND TOR HELLESETH

Abstract—**The existence of a binary [25,8,10] code is considered. It is shown that such a code must have a generator matrix of a specific form. However, all generator matrices of this form were tested and none generated a [25,8,10] code.**

I. Is There a [25,8,10] Code?

An $[n,k,d]$ code is a binary linear block code with block length n, dimension k, and minimum Hamming distance d. Define

$$n(k,d) \triangleq \min\{n \mid \text{an } [n,k,d] \text{ code exists}\}.$$

A strongly related quantity is

$$d(n,k) \triangleq \max\{d \mid \text{an } [n,k,d] \text{ code exists}\}.$$

Although general lower and upper bounds on $n(k,d)$ exist, it can sometimes be hard to determine $n(k,d)$ exactly. [1] gives a table of bounds on $d(n,k)$ for $1 \le k \le n \le 127$. For $k \le 7$, $n(k,d)$ is completely determined for any d. However, for $d \le 80$, $n(8,d)$ remains undetermined for a number of values of d [2]. In particular, the smallest unsolved case is $n(8,10)$ where it is known [1] that

$$25 \le n(8,10) \le 26. \tag{1}$$

The question to be considered in the following is:

Is there a [25,8,10] code?

II. No

Main Assumption:

$$\mathscr{C} \text{ is a } [25,8,10] \text{ code}. \tag{2}$$

Lemma 1: There is a generator matrix G' for $\mathscr{C}$ of the form

$$G' = \left(\begin{array}{cccc|c} 1 & 0 & 0 & 1 & r_1' \\ 0 & 1 & 0 & 1 & r_2' \\ 0 & 0 & 1 & 1 & r_3' \\ \hline 0 & 0 & 0 & 0 & \\ \vdots & \vdots & \vdots & \vdots & G_1 \\ 0 & 0 & 0 & 0 & \end{array}\right) \tag{3}$$

where $r_i' \in \{0,1\}^{21}$, $i = 1,2,3$, and G_1 is a generator matrix of a [21,5,10] code.

Proof: Let G'' be a generator matrix of $\mathscr{C}$, let $\mathscr{C}^{\perp}$ be the dual code of $\mathscr{C}$, and let $d^{\perp}$ be the minimum distance of $\mathscr{C}^{\perp}$. Since, from [1], $n(17,5) \ge 26$, it is clear that $d^{\perp} \le 4$. Then there exist $d^{\perp}$ columns of G'', say, $\mathrm{col}_1, \mathrm{col}_2, \cdots, \mathrm{col}_{d^{\perp}}$ such that

$$\sum_{i=1}^{d^{\perp}} \mathrm{col}_i = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Manuscript received April 25, 1989. This work was supported by the Norwegian Research Council for Science and the Humanities (NAVF).

The authors are with the Department of Informatics, Thormøhlensgt. 55, N-5008 Bergen, Norway.

IEEE Log Number 8933979.

0018-9448/90/0500-0695$01.00 ©1990 IEEE