MORE EFFICIENT SOFT DECODING OF THE GOLAY CODES

Alexander Vardy and *Yair Be'ery**

Department of Electrical Engineering – Systems,
Tel–Aviv University, Ramat–Aviv 69978, Israel
* also with the DSP Group Ltd., Givat Shmuel 51905, Israel

ABSTRACT

We present an algorithm for maximum–likelihood soft decision decoding of the binary (24,12,8) Golay code. The algorithm involves projecting the codewords of the binary Golay code onto the codewords of the (6,3,4) code over GF(4), – the hexacode. The complexity of the proposed algorithm is at most 651 real operations which is, to the best of our knowledge, less than the complexity of any algorithm ever published. Along similar lines the tetracode may be employed for decoding the ternary (12,6,6) Golay code with only 530 real operations. The proposed algorithm also implies a reduction in the number of computations required for decoding of the Leech lattice.

I. INTRODUCTION

The (24,12,8) binary extended Golay code C is certainly one of the most interesting codes known. It is an extended perfect doubly–even self–dual code. Codewords of C of weight eight, called *octads*, hold the Steiner system $S(5,8,24)$. The Golay code may be used to construct (via construction B of [8, p.142]) the Leech lattice, – an extremely dense 24 dimensional sphere packing, which has been recently utilized for the implementation of the, so called, block–coded modulation techniques for band–limited channels [5,10]. A faster algorithm for the soft decision decoding of the binary Golay code generally implies a faster algorithm for the decoding of the Leech lattice as well. Thus, the availability of an efficient Golay decoder, especially a soft decision decoder, apparently has some practical importance. In fact such a decoder was even implemented in a special purpose VLSI hardware in [1].

For all these reasons the problem of maximum–likelihood soft decision decoding of the binary Golay code was intensively investigated in the last few years. In 1986 Conway and Sloane [7] published a decoding algorithm which requires 1614 operations and in the same year Be'ery and Snyders [3] have proposed an algorithm with a worst case complexity of 1551 operations. The recent decoding algorithm of Forney [10] requires only 1351 operations. Yet, using the terminology of Forney [12], the "world record" in the decoding of the binary Golay code belongs to Be'ery and Snyders [4,18] and it stands on 827 operations at most. In this paper we claim a more efficient algorithm which requires at most 651 operations.

To be precise we note that, in compliance with the convention of [2–4,7,10–12,18], the complexity of decoding is measured herein in terms of the total number of real additions and comparisons. The memory addressing, negation and logic operations are neglected though none of these is allowed to be excessive. An effort has been made to evaluate all the algorithms in a uniform manner. The figures cited above follow those of [18]. As Conway and Sloane [7] say use these figures for comparison only.

In [9] Curtis proposed the Miracle Octad Generator (MOG) as an efficient means for locating the octad of $S(5,8,24)$ which contains 5 given points. Conway [6] has developed this idea further by introducing the *hexacode*. Conway and Sloane [8] and Pless [15] has shown how the hexacode may be used to enable hard decision decoding of the binary Golay code by hand. In the sequel we shall employ the hexacode for the soft decision decoding of the binary Golay code. In Section II we briefly outline the relation between the hexacode and the binary Golay code. For a detailed treatment see [8, chapter 11]. The decoding algorithm is presented in Section III. More efficient decoding of the ternary Golay code and of the Leech lattice is discussed in Section IV.

II. PRELIMINARIES

The hexacode H is the unique (up to a monomial permutation of coordinates) (6,3,4) linear code over $GF(4)$. Recall that the elements of $GF(4)$, – $0,1,\omega,\bar{\omega}$, hereafter called *characters*, satisfy: $\bar{\omega} = \omega^2$, $\bar{\omega}^2 = \omega$ and $\bar{\omega} = 1+\omega$. Following [15] we take

$$\begin{bmatrix} 1 & 0 & 0 & 1 & \bar{\omega} & \omega \\ 0 & 1 & 0 & 1 & \omega & \bar{\omega} \\ 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

as a generator matrix of H. The relation between H and C is as follows. We shall represent binary vectors of length 24 by 4×6 arrays with entries from $GF(2)$. A row or a column of such array is called *odd* or *even* according as it contains an odd or even number of nonzeros. If we label the four rows with distinct characters the inner product of each column of the array with the row labels will produce a character, called the *projection* of this column. The column is then said to be an *interpretation* of this character. Thus each character has four different interpretations: two odd and two even. The 16 possible interpretations of the four characters are as follows

1 0	0 1	0 1	0 1
0 1	1 0	0 1	0 1
0 1	0 1	1 0	0 1
0 1	0 1	0 1	1 0
0	1	ω	$\bar{\omega}$

odd interpretations

0 1	1 0	1 0	1 0
0 1	1 0	0 1	0 1
0 1	0 1	1 0	0 1
0 1	0 1	0 1	1 0
0	1	ω	$\bar{\omega}$

even interpretations

By taking the projection of each of the six columns we may project binary vectors of length 24 onto quaternary vectors of length 6. The equivalence of the following definition to other definitions of the (24,12,8) Golay code is proved in detail in [15]. The main idea of the proof is to show that the parameters of C, as defined below, are indeed (24,12,8) and then employ the uniqueness of the binary Golay code [14].

Definition. *The code C is the set of all the $4{\times}6$ arrays with elements from $GF(2)$, which satisfy the following conditions:*

- (i). *The parity of all the columns is the same, i.e. all the columns are either even or odd.*
- (ii). *The parity of the columns equals the parity of the top row.*
- (iii). *The projection is in the hexacode.*

□

For instance the following arrays are codewords of C.

$$\begin{matrix} 0 \\ 1 \\ \omega \\ \bar{\omega} \end{matrix} \begin{bmatrix} 1\,0\,0\,1\,0\,0 \\ 1\,0\,0\,1\,1\,1 \\ 1\,1\,0\,0\,0\,1 \\ 1\,1\,0\,0\,1\,0 \end{bmatrix} \qquad \begin{matrix} 0 \\ 1 \\ \omega \\ \bar{\omega} \end{matrix} \begin{bmatrix} 1\,0\,1\,1\,0\,0 \\ 1\,0\,0\,0\,1\,1 \\ 1\,1\,1\,0\,1\,0 \\ 0\,0\,1\,0\,1\,0 \end{bmatrix}$$

$$0\,1\,0\,1\,\omega\,\bar{\omega} \qquad\qquad \bar{\omega}\,\omega\,1\,0\,0\,1$$

III. THE DECODING ALGORITHM

Assume that a codeword of C is transmitted through a binary channel with output alphabet $\mathbb{R}$ characterized by the transition probability densities (or in case of a discrete channel, transition probabilities) $f_j(v) = f(v/j)$, where $j \in GF(2)$ and $v \in \mathbb{R}$. Let the vector $\underline{v} = (v_1, v_2, \ldots v_{24})$ be observed at the output. Maximum likelihood soft decision decoding consists of finding a codeword $\underline{c} = (c_1, c_2, \ldots c_{24}) \in C$ which maximizes $P(\underline{v}/\underline{c})$, that is maximizes the *a posteriori* probability $P(\underline{c}/\underline{v})$, provided that $P(\underline{c}) = 2^{-12}$ for all codewords of C. As shown in [3] on a random noise channel one may as well search a codeword which maximizes the *metric* $M(\underline{c})$ given by

$$M(\underline{c}) = \sum_{i=1}^{24} (-1)^{c_i} \mu_i \tag{1}$$

where $\mu_i = \log\,[f_0(v_i)/f_1(v_i)]$ is the *confidence value* of the i–th bit. In case of the additive white Gaussian noise (AWGN) channel one may set [3] $\mu_i = v_i$. The 24 real values $\mu_1, \mu_2, \ldots \mu_{24}$ are the input to our decoder. The output of the decoder is the codeword $\hat{\underline{c}} \in C$ which maximizes (1). The decoding algorithm will be described in five steps.

1. *Computing the confidence values of the characters*

For each of the six coordinates of H, i.e. j = 1,2,3,4,5,6 and for each character we shall compute the *confidence value of the even interpretation* $\mu_j^e(x)$ and the *confidence value of the odd interpretation* $\mu_j^o(x)$. The confidence values $\mu_1^e(x)$ and $\mu_1^o(x)$ are defined as follows

$$\mu_1^e(0) = |\mu_1+\mu_2+\mu_3+\mu_4| \qquad \mu_1^o(0) = |\mu_1-\mu_2-\mu_3-\mu_4|$$

$$\mu_1^e(1) = |\mu_1+\mu_2-\mu_3-\mu_4| \qquad \mu_1^o(1) = |\mu_1-\mu_2+\mu_3+\mu_4|$$

$$\mu_1^e(\omega) = |\mu_1-\mu_2+\mu_3-\mu_4| \qquad \mu_1^o(\omega) = |\mu_1+\mu_2-\mu_3+\mu_4|$$

$$\mu_1^e(\bar{\omega}) = |\mu_1-\mu_2-\mu_3+\mu_4| \qquad \mu_1^o(\bar{\omega}) = |\mu_1+\mu_2+\mu_3-\mu_4|$$

and the confidence values of the remaining 5 coordinates are defined similarly. Note that interpretations of the same character which have the same parity are complements of each other. For instance if $\mu_1+\mu_2+\mu_3+\mu_4 \geq 0$ then the confidence value of the interpretation $(0000)^t$ for the character 0 is $\mu_1^e(0)$, and the confidence value of the interpretation $(1111)^t$ is $-\mu_1^e(0)$. Thus the sign of $\mu_1+\mu_2+\mu_3+\mu_4$ determines which of the two even interpretations of the character 0 is the *preferable interpretation.*

Complexity: Using the appropriate Gray code [16], such as

$$\begin{array}{l} 1.\ + + + + \\ 2.\ - + + + \\ 3.\ - - + + \\ 4.\ + - + + \\ 5.\ + - - + \\ 6.\ - - - + \\ 7.\ - + - + \\ 8.\ + + - + \end{array} \tag{2}$$

the complexity of this step is 10 operations for each coordinate and altogether 60 operations.

2. *Sorting the confidence values of the characters*

For each of the six coordinates $j = 1,2,3,4,5,6$ we shall sort the confidence values of even and odd interpretations. Thus for example, if $\mu_1 = 0.32$, $\mu_2 = -0.25$, $\mu_3 = 0.67$ and $\mu_4 = 0.11$, we will create the following two ordered lists for the first coordinate,

$$\mu_1^e(\omega) \geq \mu_1^e(0) \geq \mu_1^e(1) \geq \mu_1^e(\bar{\omega})\ ;$$

$$\mu_1^o(1) \geq \mu_1^o(\bar{\omega}) \geq \mu_1^o(\omega) \geq \mu_1^o(0)\ .$$

The confidence values of the other 5 coordinates are sorted similarly.

Complexity: The difference between the confidence values of any two characters may be expressed in the form $\pm\, 2\mu_{i1} \pm 2\mu_{i2}$, where i_1, i_2 belong to $\{1,2,3,4\}$ for the first coordinate, and to the corresponding 4–set for the other coordinates. Thus all we have to do for the first coordinate, for instance, is to sort the absolute values $|\mu_1|, |\mu_2|, |\mu_3|, |\mu_4|$. As is well known [13] the number of comparisons required to sort four real values is 5. However, if the Gray code at step 1 is chosen appropriately this may be achieved with only 3 comparisons. Note that the Gray code of (2) is cyclic, i.e. we can start the computation at any row. In case of the example above if we start, for instance, at row 8 we can conclude that $|\mu_1| > |\mu_2|$ and $|\mu_3| > |\mu_4|$. It follows that the complexity of the second step is at most 3 operations for each coordinate and altogether at most 18 operations (at least 12).

3. *Computing the confidence values of the blocks*

If $\underline{x} = (x_1, x_3, x_3, x_4, x_5, x_6)$ is any vector in $GF(4)^6$ the sets $\{x_1, x_2\}$, $\{x_3, x_4\}$ and $\{x_5, x_6\}$ are said to be the *blocks* of $\underline{x}$. For each of the 4096 vectors of $GF(4)^6$ and for each of the three blocks we compute the *confidence value of the sum* in even interpretation $S_j^e(x_1, x_2)$ and in odd interpretation $S_j^o(x_1, x_2)$, as well as the *confidence value of the difference* in even interpretation $D_j^e(x_1, x_2)$ and in odd interpretation $D_j^o(x_1, x_2)$, where $j = 1,2,3$. For the first block the confidence values of the sum and the difference are defined by

$$
\begin{aligned}
S_1^e(x_1,x_2) &= \mu_1^e(x_1) + \mu_2^e(x_2) \\
S_1^o(x_1,x_2) &= \mu_1^o(x_1) + \mu_2^o(x_2) \\
D_1^e(x_1,x_2) &= |\mu_1^e(x_1) - \mu_2^e(x_2)| \\
D_1^o(x_1,x_2) &= |\mu_1^o(x_1) - \mu_2^o(x_2)|
\end{aligned} \tag{3}
$$

while for the other two blocks the corresponding confidence values are defined similarly.

Complexity: Computing each of the confidence values defined in (3) requires 16 operations according to the 16 possibilities of choosing $\{x_1,x_2\}$. Therefore the complexity of this step is 64 operations for each block and altogether 192 operations.

The next two steps require some explanation. Consider for instance the following hexacodeword $(0101\omega\bar{\omega})$. We shall construct a 4×6 array with entries from $GF(2)$ by taking the preferable even interpretation for each of the six characters of $(0101\omega\bar{\omega})$. Now if the top row of this array has even parity, condition (ii) is satisfied and the array corresponds to a codeword $\underline{c} \in C$ with a metric

$$M(\underline{c}) = S_1^e(0,1) + S_2^e(0,1) + S_3^e(\omega,\bar{\omega})$$

However if the top row of the array is odd, condition (ii) is violated and to obtain a codeword $\underline{c} \in C$ we must *complement* one of the characters, i.e. instead of the preferable even interpretation for this character we have to take its complement. Indeed, to achieve the highest possible metric we should complement the least reliable character, namely the one with the lowest confidence value (this is exactly the Wagner rule of [17]). For instance if $\mu_5{}^e(\omega) \le \mu_2{}^e(1) \le \dots \le \mu_3{}^e(0)$, the character in the fifth coordinate should be complemented and the metric $M(\underline{c})$ is given by

$$M(\underline{c}) = S_1^e(0,1) + S_2^e(0,1) + D_3^e(\omega,\bar{\omega})$$

The next step enables us to determine which of the six characters is to be complemented in each codeword of H if condition (ii) is violated.

4. *Merging the confidence values of the characters*

In each of the 64 hexacodewords we shall find the least reliable character in even interpretation and in odd interpretation, by means of the appropriate sorting of

$$\{\mu_1^e(0),\mu_1^e(1),\mu_1^e(\omega),\mu_1^e(\bar{\omega}); \dots ; \mu_6^e(0),\mu_6^e(1),\mu_6^e(\omega),\mu_6^e(\bar{\omega})\} \tag{4a}$$

$$\{\mu_1^o(0),\mu_1^o(1),\mu_1^o(\omega),\mu_1^o(\bar{\omega}); \dots ; \mu_6^o(0),\mu_6^o(1),\mu_6^o(\omega),\mu_6^o(\bar{\omega})\} \tag{4b}$$

Complexity: From the second step we have the sorting of the confidence values in each coordinate. Furthermore, computation of $D_1{}^e(x_1,x_2)$ at the third step is equivalent to comparison between $\mu_1{}^e(x_1)$ and $\mu_2{}^e(x_2)$. Hence after the third step we have the sorting of the eight confidence values in each block. Evidently the character which has the highest confidence value in its block will never be complemented. Now let a,b,c be the characters which have the second large confidence value in the three blocks, and assume that the confidence value of c is lower than that of a and b. Then the characters a and b will never be complemented as well. Thus the complexity of the required sort of (4a) is 2 comparisons to determine the minimum among the confidence values of a,b,c plus another $(6+6-1) + (12+7-1) = 29$ comparisons required for merging of the $(6+6+7)$ pre–sorted confidence values of the characters in the three blocks, using the technique of two–way merge [13]. Altogether the complexity of the fourth step is $2(2+29) = 62$ operations.

5. *Maximizing the metric with respect to the hexacode*

For each of the 64 hexacodewords $\underline{x} = (x_1,x_2,x_3,x_4,x_5,x_6) \in H$ we shall compute the metric of $\underline{x}$ in even interpretation and in odd interpretation, according to

$$
\begin{aligned}
M^e(\underline{x}) &= S_1^e(x_1,x_2)+S_2^e(x_3,x_4)+S_3^e(x_5,x_6) \\
M^o(\underline{x}) &= S_1^o(x_1,x_2)+S_2^o(x_3,x_4)+S_3^o(x_5,x_6)
\end{aligned}
$$

provided that the preferable interpretations of the characters of $\underline{x}$ satisfy condition (ii). If condition (ii) is violated and the least reliable character of $\underline{x}$ belongs to the j–th block, we shall replace $S_j{}^e(x_1,x_2)$ by $D_j{}^e(x_1,x_2)$ in the computation of $M^e(\underline{x})$ and/or $S_j{}^o(x_1,x_2)$ by $D_j{}^o(x_1,x_2)$ in the computation of $M^o(\underline{x})$. Among all the codewords of H we choose the codeword $\hat{\underline{x}}$ which has the highest metric in either interpretation, and decode to the codeword $\hat{\underline{c}} \in C$ whose projection is $\hat{\underline{x}}$, using the preferable interpretation for all but possibly one of the six characters.

Complexity: In a straightforward manner this step requires $2\cdot 64\cdot 2 = 256$ additions and 127 comparisons. There exists, however, a more efficient technique which will enable us to save at least 64 operations. We partition H into 16 disjoint sets of four codewords each, so that all the codewords in the same set share a common block. For instance consider the following partition

$$
\left\{\begin{matrix} 0\,0 & 0\,0\,0\,0 \\ 0\,0 & 1\,1\,1\,1 \\ 0\,0 & \omega\,\omega\,\omega\,\omega \\ 0\,0 & \bar{\omega}\,\bar{\omega}\,\bar{\omega}\,\bar{\omega} \end{matrix}\right\} ;
\left\{\begin{matrix} 0\,1 & 0\,1\,\omega\,\bar{\omega} \\ 0\,1 & 1\,0\,\bar{\omega}\,\omega \\ 0\,1 & \omega\,\bar{\omega}\,0\,1 \\ 0\,1 & \bar{\omega}\,\omega\,1\,0 \end{matrix}\right\} ; \quad \cdots \quad ;
\left\{\begin{matrix} \bar{\omega}\,\bar{\omega} & 0\,0\,\bar{\omega}\,\bar{\omega} \\ \bar{\omega}\,\bar{\omega} & 1\,1\,\omega\,\omega \\ \bar{\omega}\,\bar{\omega} & \omega\,\omega\,1\,1 \\ \bar{\omega}\,\bar{\omega} & \bar{\omega}\,\bar{\omega}\,0\,0 \end{matrix}\right\}
$$

Finding the codeword with the highest metric in each of the sets above requires at most 9 operations (at least 8 operations) for each interpretation. Hence the complexity of the fifth step is at most $2\cdot 16\cdot 9 + 31 = 319$ real operations. ■

Thus the total number of real operations required in our algorithm is $60+18+192+62+319 = 651$ in the worst case, as compared to the best decoding algorithm presently known [18] which requires 827 operations. Note that the five steps are essentially independent and therefore enable pipelining. Furthermore, each of the five steps offers an appreciable amount of inherent concurrence. These two facts may be employed for an efficient hardware implementation of the proposed decoder.

IV. FURTHER RESULTS

The *tetracode* is the unique, up to monomial equivalence, (4,2,3) code over $GF(3)$ generated by

$$\begin{bmatrix} 1\,0\,1\,2 \\ 0\,1\,1\,1 \end{bmatrix}$$

In [6,8] Conway has investigated the relation between the tetracode and the (12,6,6) ternary Golay code. Pless [15] has employed the tetracode for the hard decision decoding of the ternary Golay code. It turns out that the tetracode may be used for efficient soft decision decoding as well. Thus decoding the ternary Golay code by means of projecting its 729 codewords onto the nine codewords of the tetracode requires only 530 real operations in the worst case; as compared to 656 real operations of [18] and 1061 operations of [11]. This suggests that the approach employed herein for the decoding of the binary Golay code apparently has a natural generalization, i.e. efficient soft decision decoding of certain codes might be obtained by projecting them onto shorter codes with less codewords.

It is noteworthy that the above reduction of the complexity of the maximum likelihood decoding of the binary Golay code has immediate consequences for the decoding of the Leech lattice as well. Thus for instance, Forney [12] employs the algorithm of [18] for a bounded–distance decoding of the Leech lattice by means of about 2000 operations. Substituting the proposed algorithm into that of Forney [12] enables bounded–distance decoding of the Leech lattice with only about 1500 operations. In [2] Be'ery, Shahar and Snyders develop an optimal algorithm for the decoding of the Leech lattice (to the best of our knowledge the most efficient algorithm ever published) which requires about 6000 operations. The algorithm presented herein implies that the computational complexity of [2] may be reduced by approximately the same factor.

ACKNOWLEDGEMENT

This paper is a consequence of the last in a series of lectures on coding theory that Prof. Vera Pless gave at the Technion. During this lecture she has raised the question: whether the hexacode could be employed for soft decision decoding of the binary Golay code. In a sense the derivation above is simply an answer to this question. The authors are grateful to Vera Pless for bringing the matter to their attention. Alexander Vardy wishes to thank Hagit Itzkowitz for her invaluable help.

REFERENCES

[1] A.D. Abbaszadeh and C.K. Rushforth, "VLSI implementation of a maximum likelihood decoder for the Golay(24,12) code," *IEEE J. Sel. Areas Comm.*, vol.6, pp. 558–565, 1988.

[2] Y. Be'ery, B. Shahar, and J. Snyders, "Fast decoding of the Leech lattice," *IEEE J. Sel. Areas Comm.*, vol.7, pp. 959–967, 1989.

[3] Y. Be'ery and J. Snyders, "Optimal soft decision block decoders based on Fast Hadamard Transform," *IEEE Trans. Inform. Theory*, vol.32, pp. 355–364, 1986.

[4] Y. Be'ery and J. Snyders, "New methods for soft decision decoding of the Golay (24,12) code," in *Proc. IEEE 15th Conf. Electr. Eng. in Israel*, Tel–Aviv, Israel, 1987.

[5] A.R. Calderbank and N.J.A. Sloane, "New trellis codes based on lattices and cosets," *IEEE Trans. Inform. Theory*, vol.33, pp. 177–195, 1987.

[6] J.H. Conway, "Hexacode and tetracode – MOG and MINIMOG," in M.D. Atkinson, ed. *Computational Group Theory*, Academic Press, New York, pp. 359–365, 1984.

[7] J.H. Conway and N.J.A. Sloane, "Soft decoding techniques for codes and lattices, including the Golay code and the Leech lattice," *IEEE Trans. Inform. Theory*, vol.32, pp. 41–50, 1986.

[8] J.H. Conway and N.J.A. Sloane, *Sphere Packings, Lattices and Groups*, Springer–Verlag, New York, 1988.

[9] R.T. Curtis, "A new combinatorial approach to M_{24}," *Math. Proc. Camb. Phil. Soc.*, vol.79, pp. 25–41, 1976.

[10] G.D. Forney, Jr., "Coset codes II: Binary lattices and related codes," *IEEE Trans. Inform. Theory*, vol.34, pp. 1152–1187, 1988.

[11] G.D. Forney, Jr., "Coset codes III: Ternary codes, lattices and trellis codes," *IEEE Trans. Inform. Theory*, to appear.

[12] G.D. Forney, Jr., "A bounded distance decoding algorithm for the Leech lattice, with generalizations," *IEEE Trans. Inform. Theory*, vol.35, pp. 906–909, 1989.

[13] D.E. Knuth, *The Art of Computer Programming*, vol.3: *Sorting and Searching*, Addison–Wesley, Reading, MA., 1973.

[14] V.S. Pless, "On the uniqueness of the Golay codes," *J. Comb. Theory*, vol.5, pp. 215–228, 1968.

[15] V.S. Pless, "Decoding the Golay Codes," *IEEE Trans. Inform. Theory*, vol.32, pp. 561–567, 1986.

[16] E.M. Reingold, J. Nievergelt, and N. Deo, *Combinatorial Algorithms: Theory and Practice*, Prentice–Hall, Englewood Cliffs, NJ., 1977.

[17] R.A. Silverman and M. Balser, "Coding for a constant data rate source," *IRE Trans. Inform. Theory*, vol.4, pp. 50–63, 1954.

[18] J. Snyders and Y. Be'ery, "Maximum likelihood soft decoding of binary block codes and decoders for the Golay codes," *IEEE Trans. Inform. Theory*, vol.35, No.5, 1989.