High-Speed Decoding of the Binary Golay Code

H. P. Lee[*1], C. H. Chang[1], S. I. Chu[2]

[1] Department of Computer Science and Information Engineering,
Fortune Institute of Technology,
Kaohsiung 83160, Taiwan
*hpl@fotech.edu.tw
[2] Department of Electronic Engineering,
National Kaohsiung University of Applied Sciences,
Kaohsiung 807, Taiwan

ABSTRACT
Recently, some table-lookup decoding algorithms (TLDAs) have been used to correct the binary Golay code. This paper proposes an efficient high-speed TLDA called *message-syndrome decoding algorithm* (MSDA) by using the message syndrome to correct the binary systematic Golay code. The proposed MSDA is based on the novel message-syndrome lookup table (MSLT). The MSLT merely consists of 12 candidate syndromes and its memory size is significantly smaller than other existing lookup tables. Computer software simulation results show that the decoding speed of the proposed MSDA is superior to other proposed TLDAs.

Keywords: Golay code, weight, syndrome, error pattern

1. Introduction

The well-known binary Golay code, also called the *binary* (23, 12, 7) *quadratic residue* (QR) code [2], was first discovered by Golay [1] in 1949. It is a very useful perfect linear error-correcting code; particularly, it has been used in the past decades for a variety of applications involving a parity bit being added to each word to yield a half-rate code called the *binary* (24, 12, 8) *extended Golay code*. One of its most interesting applications is the provision of error control in the Voyager missions [2]. The Golay code can allow the correction of up to $t = \lfloor (d-1)/2 \rfloor = \lfloor (7-1)/2 \rfloor = 3$ errors, where $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x, t is the error-correcting capability, and d is the minimum Hamming distance of the code.

In the past few decades, several decoding techniques have been developed to decode the binary Golay codes, for example, the famous algebraic decoding algorithm (ADA) developed by Elia [3], the notable shift-search algorithm proposed by Reed et al. [4] after that, and the inverse-free Berlekamp–Massey algorithm used to obtain the error-locator polynomial by Chen et al. [5]. The above is some representative literature on ADAs used to correct Golay codes.

Recently, the TLDAs given in [6-9] have acquired a prominent role in error-correcting decoding. A TLDA with a refined lookup table (RLT) has been proposed by Lin et al. [8]; the RLT only needs 168 bytes memory size. The memory size of the RLT is the smallest memory size required to correct the Golay code until now. In general, the TLDA makes considerably faster decoding speeds possible for the cyclic codes, in comparison with the ADA. However, the TLDA requires an extra memory space in the decoder chip or software and it increases the decoding cost rapidly when the code length is large. Although the ADA does not need any lookup table, it requires a large number of complicated computations in the finite field. These complicated computations will degrade the decoding performance and lead to a serious decoding delay as the code length increases [10].

Chen et al. [7] proposed the fast lookup table decoding algorithm (FLTDA) with a lookup table of 1.35K bytes by using the shift-search algorithm [4] to decode the Golay code. The famous syndrome decoding algorithm (SDA) with a reduced-size lookup table (RSLT) given in [2, p119] requires 89 syndromes and their corresponding error patterns, whereas the RSLT requires 445 bytes of memory.

Lin et al. [8] proposed the syndrome-weight decoding algorithm (SWDA) with RLT which needs 168 bytes of memory, and the simulation result shows that the decoding speed is approximately 9.7 times faster than the improved Elia's decoding algorithm (EDA).

In this paper, an efficient high-speed MSDA with a MSLT is proposed to decode the binary systematic Golay code. The novel MSLT consists only of 12 candidate syndromes which have one error in the message part of the received vector, and it merely requires 24 bytes of memory. Software simulation results show that the average decoding speed is the fastest among the TLDAs as mentioned above. In the four-error soft-decision decoder in [11], the improved EDA is the role of the hard-decision decoder. We substituted the proposed MSDA for the improved EDA in the four-error soft-decision decoder. Apparently, the decoding speed of the soft-decision decoder was faster than before.

The remainder of this paper is organized as follows: The background of the binary systematic Golay code is briefly reviewed in Section 2. Some related TLDAs and related theorems are briefly introduced in Section 3. The proposed MSDA is presented in Section 4. Computer simulation results are given in Section 5. Finally, this paper concludes with a brief summary in Section 6.

2. Background of the binary systematic Golay code

The binary (23, 12, 7) Golay code can be defined as a linear cyclic code or QR code [2]. For the binary Golay code over finite field $GF(2^{11})$, its quadratic residue set is given by

$$\begin{aligned} Q_{23} &= \{i \mid i \equiv j^2 \bmod 23 \text{ for } 1 \le j \le 22\} \\ &= \{1, 2, 3, 4, 6, 8, 9, 12, 13, 16, 18\}. \end{aligned} \quad (1)$$

The generator polynomial $g(x)$ of the binary Golay code is defined by

$$\begin{aligned} g(x) &= \prod_{i \in Q_{23}} (x - \beta^i) \\ &= x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1, \end{aligned} \quad (2)$$

where β is a primitive 23rd root of unity in $GF(2^{11})$. Let the codeword polynomial $C(x) = \sum_{i=0}^{n-1} C_i x^i$, where $C_i \in GF(2)$ for $0 \le i \le 22$, and the message polynomial $m(x) = \sum_{i=0}^{k-1} m_i x^i$, where $k = 12$ is the message length and $m_i \in GF(2)$ for $0 \le i \le 11$. $C(x)$ is a multiple of $g(x)$, namely, $C(x) = m(x)g(x)$. Also, let the product $m(x)x^{11}$ divide by $g(x)$, then one has $m(x)x^{11} = q(x)g(x) + d(x)$. Multiplying both sides of last expression by x^{12} and using $x^{23} = 1$, one yields $d(x)x^{12} + m(x) = (q(x)x^{12})g(x)$. The term $(d(x)x^{12} + m(x))$, which is a multiple of $g(x)$, is a systematic codeword given by

$$c(x) = d(x)x^{12} + m(x) = p(x) + m(x), \quad (3)$$

where $p(x) = \sum_{i=12}^{22} p_i x^i$ denotes the parity-check polynomial [12] of a codeword and $p_i \in GF(2)$ for $12 \le i \le 22$.

The binary systematic Golay code is practically applied in the standard of the communication system for automatic link establishment (ALE) by the United States Department of Defense [13]. Now, let a 12-bit message be encoded into a 23-bit systematic codeword and be transmitted through a noisy channel to obtain a received word of the form $r(x) = c(x) + e(x)$. $e(x) = \sum_{i=0}^{22} e_i x^i$ is the error pattern polynomial, where $e_i \in GF(2)$ for $0 \le i \le 22$. The syndromes of the code are defined by

$$\begin{aligned} s_i &= r(\beta^i) = c(\beta^i) + e(\beta^i) \\ &= e(\beta^i) = \sum_{j=0}^{22} e_j (\beta^i)^j, \end{aligned} \quad (4)$$

where $i \in Q_{23}$.

To simplify the polynomial expressions above, the message, codeword, error pattern, received word, and syndrome polynomials can be expressed as the binary vector forms $\mathbf{m} = (m_{11} \ldots m_1\ m_0)$, $\mathbf{c} = (c_{22} \ldots c_1\ c_0)$, $\mathbf{e} = (e_{22} \ldots e_1\ e_0)$, $\mathbf{r} = \mathbf{c} + \mathbf{e} = (r_{22} \ldots r_1\ r_0)$, and $\mathbf{s} = (s_{10} \ldots s_1\ s_0)$, respectively. For the binary systematic Golay code, it follows from [2, p85] that the systematic generator matrix $\mathbf{G}$ can be expressed as follows:

$$\mathbf{G} = \left[\mathbf{P}\middle|\mathbf{I}_{12}\right]_{12\times 23} = \left[\begin{array}{c|c} 10101110001 & \\ 11111001001 & \\ 11010010101 & \\ 11000111011 & \\ 11001101100 & \\ 01100110110 & \mathbf{I}_{12} \\ 00110011011 & \\ 10110111100 & \\ 01011011110 & \\ 00101101111 & \\ 10111000110 & \\ 01011100011 & \end{array}\right]_{12\times 23}, \quad (5)$$

where $\mathbf{P}$ and $\mathbf{I}_{12}$ denote the 12×11 matrix and the 12×12 identity matrix, respectively. The systematic vector form of the codeword can be obtained by

$$\mathbf{c} = \mathbf{mG} = (p_{22} \ldots p_{13}\ p_{12}\middle|\ m_{11} \ldots m_1\ m_0). \quad (6)$$

The parity-check polynomial $h(x) = x^{12} + x^{10} + x^7 + x^4 + x^3 + x^2 + x + 1$ is a factor of $x^{23} - 1$. The 11×23 parity-check matrix $\mathbf{H}$ can be expressed as follows:

$$\mathbf{H} = \left[\mathbf{I}_{11}\middle|\mathbf{P}^T\right]_{11\times 23}, \quad (7)$$

where $\mathbf{P}^T$ is a 11×12 transpose matrix of $\mathbf{P}$ and $\mathbf{I}_{11}$ is a 11×11 identity matrix.

The $\mathbf{H}^T$ denotes the 23 × 11 transpose matrix of $\mathbf{H}$; that is, $\mathbf{H}^T = \left[\frac{\mathbf{I}_{11}}{\mathbf{P}}\right]_{23\times 11}$. The vector form of the syndrome can be defined by

$$\mathbf{s} = \mathbf{rH}^T. \quad (8)$$

The following lemma given in [14] shows that the mapping between the syndromes and error patterns under the error-correcting capability is one-to-one.

Lemma 1. The mapping between the syndromes of a code and the corresponding error patterns of weight ≤ t is one-to-one.

3. Related work and theorems

In this section, some related TLDAs and theorems are briefly introduced in order to develop the proposed MSDA. For the TLDA, the following definition, theorems and corollary given in [9] are needed.

Definition 1. Let the Hamming norm or weight of a binary vector $\mathbf{a} = (a_{n-1} \ldots a_1\ a_0)$ be designated by $w(\mathbf{a})$. The Hamming distance between two binary vectors $\mathbf{a}$ and $\mathbf{b}$ is defined by $d(\mathbf{a}, \mathbf{b}) = w(\mathbf{a} + \mathbf{b})$.

Theorem 1. Let $\mathbf{a} = (a_{n-1} \ldots a_1\ a_0)$ and $\mathbf{b} = (b_{n-1} \ldots b_1\ b_0)$ be two binary vectors, then

$$w(\mathbf{a} + \mathbf{b}) = w(\mathbf{a}) + w(\mathbf{b}) - 2\sum_{i=1}^{n} a_i b_i . \quad (9)$$

Corollary 1. Let $\mathbf{a}$ and $\mathbf{b}$, defined in Definition 1, be two binary vectors. If $a_i b_i = 0$ for $1 \le i \le n$, then

$$w(\mathbf{a} + \mathbf{b}) = w(\mathbf{a}) + w(\mathbf{b}). \quad (10)$$

Theorem 2. For the binary systematic (n, k, d) QR codes, let us assume that there are v errors in the received word, where $1 \le v \le t$. All v errors occur in the $n - k$ parity-check bits if and only if the weight of syndrome $w(\mathbf{s}) = v$.

As mentioned above, the error-correcting capability of the binary Golay code is t =3. The full lookup table (FLT) in the conventional table lookup decoding algorithm (CTLDA) requires $\sum_{i=1}^{3}\binom{23}{i}$ = 2,047 syndromes and their corresponding error patterns by Lemma 1. Each syndrome and error pattern needs 2 bytes and 3 bytes, respectively, to store in the FLT. Thus, the total memory size needed for the FLT is (2047 × (2 + 3)) = 10,235 bytes. However, such a large memory is less efficient in practice.

Recently, some useful TLDAs have been presented in the literature to reduce the memory size of the FLT. The LTDA [7] used the shift-search method [4] to reduce the memory size of the FLT and the binary-search method to reduce the search time in the FLT. The shift-search method deletes all $v = t$ error patterns and keeps $1 \le v \le t - 1$ error patterns.

Therefore, the LTDA needs $\sum_{i=1}^{t-1}\binom{23}{i} = \sum_{i=1}^{2}\binom{23}{i} = 276$ syndromes and their corresponding error patterns; that is, the FLT needs 276 × (2 + 3) = 1,380 bytes memory size. Nevertheless, the memory size of the FLT is still large and the simulation result shows that the average decoding time of the proposed MSDA is about 10.6 times faster that the LTDA. To develop the following TLDAs, Theorem 3 is useful to reduce the memory size of the FLT. The well-known SDA with RSLT [2, p119] has the following useful theorem of the cyclic codes to dramatically reduce the memory size of the FLT. For more detailed proof of this theorem, see [2, p118].

Theorem 3. Let $s(x)$ be the syndrome polynomial corresponding to a received polynomial $r(x)$. Also, let $r^{(1)}(x)$ be the polynomial obtained by cyclically shifting the coefficients of $r(x)$ one bit to the left. Then the remainder obtained when dividing $xs(x)$ by $g(x)$ is the syndrome $s^{(1)}(x)$ corresponding to $r^{(1)}(x)$.

Decoding the binary systematic Golay code by Theorem 3, the RSLT of the SDA only needs 1/23 memory size of the FLT, namely 10,235/23 = 445 bytes memory size [8]. Nonetheless, the memory size of the RSLT is still large and can be further reduced. Similarly, by Theorem 3, $r^{(j)}(x)$ is the polynomial obtained by cyclically shifting the coefficients of $r(x)$ j bits to the left. Next, the syndrome difference $\mathbf{s}_{di}$ is defined by $\mathbf{s}_{di} = \mathbf{s} - \mathbf{s}_i$ for $1 \le i \le N$, where $\mathbf{s}$ is the syndrome of $\mathbf{r}$, $\mathbf{s}_i$ is the syndrome in lookup table, and N is the table size. Similarly, the syndrome difference $\mathbf{s}_{di}^{(j)}$ is defined by $\mathbf{s}_{di}^{(j)} = \mathbf{s}^{(j)} - \mathbf{s}_i$ for $1 \le j \le n$, where $\mathbf{s}^{(j)}$ is the syndrome of $\mathbf{r}^{(j)}$. By using the property of the syndrome difference, the RLT of the SWDA only requires 168 bytes. However, the decoding speed is not very outstanding according to the simulation result. To overcome this problem, the MSDA is presented by using the syndrome that only one error occurred in the message part of the received word.

4. MSDA for the binary Golay code

In this section, an efficient high-speed MSDA with a MSLT for decoding the binary systematic Golay code is developed to further reduce the decoding time and the memory size of the RLT. By Theorem 2 and (8), it is obvious that at least one error is in the message bits if the weight of syndrome $w(\mathbf{s}) > 3$.

Definition 2. The message-syndrome lookup table (MSLT) is a table of all the error patterns occurred in the message part with weight $1 \le v \le \lfloor t/2 \rfloor$ together with their corresponding syndromes, where $\lfloor x \rfloor$ denotes the smallest integer less than or equal to x.

Therefore, the novel MSLT merely consists of 12 candidate syndromes which are the syndromes of having one error in the message part; that is, $\mathbf{s}_i$ is the syndrome of $\mathbf{e}_i$ which is one error located at bit i for $1 \le i \le 12$. The MSLT is listed in Table 1.

Syndrome	Syndrome
$\mathbf{s}_1$ = (01011100011)	$\mathbf{s}_2$ = (10111000110)
$\mathbf{s}_3$ = (00101101111)	$\mathbf{s}_4$ = (01011011110)
$\mathbf{s}_5$ = (10110111100)	$\mathbf{s}_6$ = (00110011011)
$\mathbf{s}_7$ = (01100110110)	$\mathbf{s}_8$ = (11001101100)
$\mathbf{s}_9$ = (11000111011)	$\mathbf{s}_{10}$ = (11010010101)
$\mathbf{s}_{11}$ = (11111001001)	$\mathbf{s}_{12}$ = (10101110001)

Table 1. MSLT for the Binary Systematic Golay Code.

To develop the proposed MSDA by using the MSLT, the following lemma and properties are necessary.

Lemma 2. For the binary Golay code, let $\mathbf{e}$ be an error pattern, and let $\mathbf{e}_p$ and $\mathbf{e}_m$ be its parity-check part and message part, respectively. Let $\mathbf{s}_p$ and $\mathbf{s}_m$ be the syndromes corresponding to $\mathbf{e}_p$ and $\mathbf{e}_m$, respectively. Assume that $w(\mathbf{e}) \le 3$, then we have

(i) $w(\mathbf{s}_p) = w(\mathbf{e}_p)$;

(ii) $w(\mathbf{s}_m) \ge d - w(\mathbf{e}_m)$, where $d = 7$;

(iii) $((\mathbf{s}_m << 12) + \mathbf{e}_m)$ is always a codeword, where "<<" denotes the logical left shift operator in programming or the extension by zeros on the right in mathematics.

Proof: By Equation (8), (i) is obvious. Again, by Equation (8), given that $((\mathbf{s}_m << 12) + \mathbf{e}_m)\mathbf{H}^T = ((\mathbf{s}_m << 12) + \mathbf{e}_m)\left[\frac{\mathbf{I}_{11}}{\mathbf{P}}\right] = (\mathbf{s}_m << 12)\left[\frac{\mathbf{I}_{11}}{\mathbf{P}}\right] + \mathbf{e}_m\left[\frac{\mathbf{I}_{11}}{\mathbf{P}}\right] = \mathbf{s}_m + \mathbf{s}_m = \mathbf{0}$, $((\mathbf{s}_m << 12) + \mathbf{e}_m)$ is thus a codeword, which proves (iii). Hence, $w((\mathbf{s}_m << 12) + \mathbf{e}_m) = w(\mathbf{s}_m) + w(\mathbf{e}_m) \ge d$, that is, $w(\mathbf{s}_m) \ge d - w(\mathbf{e}_m)$. The proof is thus completed.

By Lemma 2, if $v \le t$, then one can obtain the following two properties.

Property 1. For the binary Golay code, the weight of the syndrome difference $w(\mathbf{s}_{di})$ has the following cases:

Case 1: If $w(\mathbf{s}) \le 3$, then all three errors are in the parity-check part and no error is in the message part.

Case 2: If $w(\mathbf{s}) > 3$ and $w(\mathbf{s}_{di}) < 3$, then only 1 error is in the message part.

Case 3: If $w(\mathbf{s}) > 3$ and $w(\mathbf{s}_{di}) > 3$, than at least 2 errors are in the message part.

Next, for Case 3 of Property 1, the **r** and **s** are cyclically shifted left by 11 bits, then one obtains $\mathbf{r}^{(11)}$ and $\mathbf{s}^{(11)}$, respectively. Thus, one yields the following property.

Property 2 For the binary Golay code, if $w(\mathbf{s}) > 3$ and $w(\mathbf{s}_{di}) > 3$, then the weight of $\mathbf{s}^{(11)}$ and $\mathbf{s}_{di}^{(11)}$ have the following cases:

Case 1: If $w(\mathbf{s}^{(11)}) = 2$ or 3, then **r** has at least 2 errors in the message part and no error is in the parity-check part.

Case 2: If $w(\mathbf{s}^{(11)}) > 3$ and $w(\mathbf{s}_{di}^{(11)}) \le 2$, then **r** has at least 2 errors in the message part.

Case 3: If $w(\mathbf{s}^{(11)}) > 3$ or $w(\mathbf{s}_{di}^{(11)}) > 3$, then 2 errors are in the message part and one of them is at the position of bit 0.

By using the MSLT, the decoding process of the MSDA is as follows:

(1) Giving a received vector **r**.

(2) Compute the syndrome $\mathbf{s} = \mathbf{rH}^{\mathbf{T}}$ and its weight $w(\mathbf{s})$.

(3) If $w(\mathbf{s}) \le 3$, then return $\mathbf{c} = \mathbf{r} - (\mathbf{s} << 12)$, and go to step 13.

(4) For $1 \le i \le 12$, compute the syndrome $\mathbf{s}_{di} = \mathbf{s} - \mathbf{s}_i$ and its weight $w(\mathbf{s}_{di})$.

(5) If $w(\mathbf{s}_{di}) \le 2$, then return $\mathbf{c} = \mathbf{r} - (\mathbf{s}_{di} << 12) - (1 << (i - 1))$, and go to step 13.

(6) Cyclically shift **r** and **s** left by 11 bits obtaining $\mathbf{r}^{(11)}$ and $\mathbf{s}^{(11)}$, and then compute $w(\mathbf{s}^{(11)})$.

(7) If $w(\mathbf{s}^{(11)}) = 2$ or 3, then return $\mathbf{c} = ((\mathbf{r}^{(11)} - (\mathbf{s}^{(11)} << 12)) << 12) - ((\mathbf{r}^{(11)} - (\mathbf{s}^{(11)} << 12)) >> 11)$, and go to step 13.

(8) For $1 \le i \le 12$, compute the syndrome $\mathbf{s}_{di}^{(11)} = \mathbf{s}^{(11)} - \mathbf{s}_i$ and its weight $w(\mathbf{s}_{di}^{(11)})$.

(9) If $w(\mathbf{s}_{di}^{(11)}) = 1$ or 2, then return $\mathbf{c} = ((\mathbf{r}^{(11)} - (\mathbf{s}_{di}^{(11)} << 12) - (1 << (i - 1))) << 12) - ((\mathbf{r}^{(11)} - (\mathbf{s}_{di}^{(11)} << 12) - (1 << (i - 1))) >> 11)$, and go to step 13.

(10) Compute $\mathbf{r}' = \mathbf{r} - 1$ and $\mathbf{s}' = \mathbf{s} - \mathbf{s}_1$.

(11) For $2 \le i \le 12$, compute the syndrome $\mathbf{s}'_{di} = \mathbf{s}' - \mathbf{s}_i$ and its weight $w(\mathbf{s}'_{di})$.

(12) If $w(\mathbf{s}'_{di}) = 1$, then return $\mathbf{c} = \mathbf{r}' - (\mathbf{s}'_{di} << 12) - (1 << (i - 1))$.

(13) Stop.

An example is given below. By (6), a message vector **m** = (100000000000) is encoded into a binary systematic Golay code **c** = (10101110001100000000000). Assume that a noise vector is **e** = (00010000000000010000001), then the received word becomes **r** = **c** + **e** = (10111110001100010000001). The decoding steps proceed as follows:

(1) The received vector is **r** = (10111110001100010000001).

(2) Compute the syndrome **s** = (10000001111) and its weight $w(\mathbf{s}) = 5$.

(3) Given that $w(\mathbf{s}) > 3$, go to step 4.

(4) For $i = 1$, compute the syndrome $\mathbf{s}_{d1} = \mathbf{s} - \mathbf{s}_1$ = (11011101100) and its weight $w(\mathbf{s}_{d1}) = 7$.

(5) Given that $w(\mathbf{s}_{d1}) > 2$, go to step 4.
…repeating steps 4 and 5…

(4) For $i = 12$, compute the syndrome $\mathbf{s}_{d12} = \mathbf{s} - \mathbf{s}_{12}$ = (00101111110) and its weight $w(\mathbf{s}_{d12}) = 7$.

(5) Given that $w(\mathbf{s}_{d12}) > 2$, go to step 6.

(6) Cyclically shift **r** and **s** left by 11 bits obtaining $\mathbf{r}^{(11)}$ = (10001000000110111110 001) and $\mathbf{s}^{(11)}$ = (01101011101), and then compute $w(\mathbf{s}^{(11)}) = 7$.

(7) Given that $w(\mathbf{s}^{(11)}) \neq 2$ or 3, go to step 8.

(8) For $i = 1$, compute the syndrome $\mathbf{s}_{d1}^{(11)} = \mathbf{s}^{(11)} - \mathbf{s}_1$ = (00110111110) and its weight $w(\mathbf{s}_{d1}^{(11)}) = 7$.

(9) Given that $w(\mathbf{s}_{d1}^{(11)}) > 2$, go to step 8.
…repeating steps 8 and 9…

(8) For $i = 12$, compute the syndrome $\mathbf{s}_{d12}^{(11)} = \mathbf{s}^{(11)} - \mathbf{s}_{12}$ = (11000101100) and its weight $w(\mathbf{s}_{d12}^{(11)}) = 5$.

(9) Given that $w(\mathbf{s}_{d12}^{(11)}) > 2$, go to step 10.

(10) Compute $\mathbf{r}' = \mathbf{r} - 1$ = (10111110001100010000000) and $\mathbf{s}' = \mathbf{s} - \mathbf{s}_1$ = (11011101100).

(11) For $i = 2$, compute the syndrome $\mathbf{s}'_{d2} = \mathbf{s}' - \mathbf{s}_2$ = (01100101010) and its weight $w(\mathbf{s}'_{d2}) = 5$.

(12) Given that $w(\mathbf{s}'_{d2}) > 1$, go to step 11.
…repeating steps 11 and 12…

(11) For $i = 8$, compute the syndrome $\mathbf{s}'_{d8} = \mathbf{s}' - \mathbf{s}_8$ = (00010000000) and its weight $w(\mathbf{s}'_{d8}) = 1$.

(12) If $w(\mathbf{s}'_{d8}) = 1$, then return the correct codeword $\mathbf{c} = \mathbf{r}1 - (\mathbf{s}'_{d8} << 12) - (1 << (i - 1))$ = (10111110001100010000000) – (00010000000000000000000) – (10000000) = (10101110001100000000000). Go to step 13.

(13) Stop.

Therefore, the correct message can be retrieved from the received vector.

5. Simulation results

The proposed MSDA has been programmed in C++ language. On an Intel Q6600 PC, all codewords with up to three errors, namely $2^{12} \times \sum_{i=1}^{3} C_i^{23} = 8{,}384512$ received vectors, were created to check every possible error pattern. The detailed decoding times for decoding the binary systematic Golay code with different decoding algorithms are given in Table 2. The proposed MSDA has the fastest average decoding time and works perfectly with an average decoding time of 0.6038 μs per error pattern.

Algorithms	Number of errors			
	1	2	3	Average
Proposed MSDA	0.3396	0.4228	0.6364	0.6038
SDA with RSLT	0.7332	0.8612	0.8876	0.8705
FLTDA with table	0.1774	0.1813	7.406	6.419
SWDA with RLT	4.651	5.957	7.551	7.321
Improved EDA	16.32	27.25	27.33	26.86

Table 2. Computer Decoding Time for Decoding the Binary Golay Code (in μs).

From Table 2, the proposed MSDA has the fastest average decoding time and is about 44.5 times faster than the improved EDA. Furthermore, we substituted the proposed MSDA for the improved EDA in the four-error soft-decision decoder [11]. Apparently, the decoding speed of the soft-decision decoder was about 44.5 times faster than before. The decoding time of the two hard-decision decoders with respect to the SNR values is shown in Figure 1.

6. Conclusions

This paper presented a simple and more efficient decoding algorithm for decoding the binary systematic Golay code. The key ideas of the MSDA are based on the properties of cyclic codes, the one-error syndrome in the message part, the weight of syndrome, the weight of syndrome

difference, Property 1, and Property 2. The MSLT of the MSDA merely requires a very small memory size of 24 bytes. These facts lead to a substantial improvement in decoding the binary systematic Golay code. Simulation results show that, following the new decoding strategies, the proposed MSDA achieves the best efficiency. Hence, it is naturally suitable and proper for DSP or embedded system software implementation.

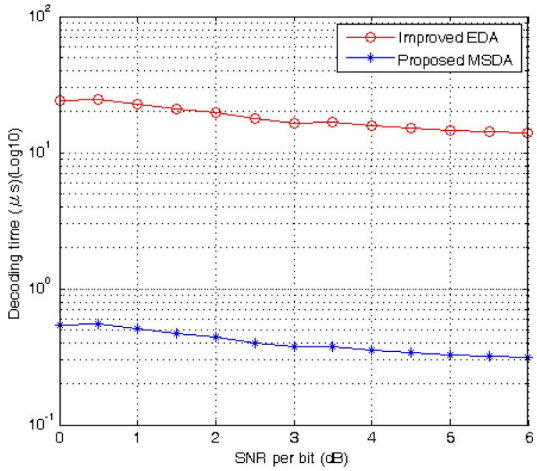

Figure 1. Decoding Time of Two Hard-Decision Decoders at SNR of 0~6 dB.

References

[1] M. Golay, “Notes on digital coding,” Proc. IRE, vol. 37, p. 657, 1949.

[2] S. Wicker, "Error Control Systems for Digital Communication and Storage”, Prentice Hall: Upper Saddle River, NJ, 1995.

[3] M. Elia, “Algebraic decoding of the (23, 12, 7) Golay code,” IEEE Trans Inf Theory, vol. 33, no. 1, pp. 150-151, 1987.

[4] I. S. Reed et al., “Decoding the (24, 12, 8) Golay code,” IEE P-COMPUT DIG T, vol. 137, no. 3, pp. 202-206, 1990.

[5] Y. H. Chen et al., Algebraic decoding of quadratic residue codes using Berlekamp-Massey algorithm, J Inf Sci Eng, vol. 23, no. 1, 2007, pp. 127-145.

[6] H. Chang et al., “A weight method of decoding the (23, 12, 7) Golay code Using Reduced Lookup Table,” 2008 International Conference On Communication, Circuits and Systems (ICCCAS 2008), Xiamen, China, 2008, pp. 1-5.

[7] Y. H. Chen et al., “Efficient decoding of systematic (23, 12, 7) and (41, 21, 9) quadratic residue codes,” J Inf Sci Eng, vol. 26, no. 5, pp. 1831-1843, 2010.

[8] T. C. Lin et al., “On the decoding of the (24, 12, 8) Golay code,” Inf Sci, vol. 180, no. 23, pp. 4729-4736, 2010.

[9] T. C. Lin et al., “High speed decoding of the binary (47, 24, 11) quadratic residue code,” Inf Sci, vol. 180, no. 20, pp. 4060-4068, 2010.

[10] S. Lin and E. J. Jr Weldon, “Long BCH codes are bad,” Inf Contr, vol. 11, no. 4, pp. 445-451, 1967.

[11] S. I. Chu et al., “Fast decoding of the (23, 12, 7) Golay code with four-error-correcting capability,” Eur Trans Telecomm, vol, 22, no. 7, pp. 388-395, 2011.

[12] C. A. Vázquez-Fernández and G. Vega-Hernández, “On the Weight Distribution of the Dual of some Cyclic Codes with Two Non Conjugated Zeros,” J Appl Res Technol, vol.9, no.1, pp. 36-48, 2011.

[13] MIL-STD-188/141B. Department of Defense Interface Standard: Interoperability and Performance Standards for Medium and High Frequency Radio Systems. http://www.everyspec.com/MIL-STD/MIL-STD+(0100+-+0299)/MIL_STD_188_141B_1703.

[14] I. S Reed et al., “The algebraic decoding of the (41, 21, 9) quadratic residue code,” IEEE Trans Inf Theory, vol. 38, no. 3, pp. 974-986, 1992.