Lee, H. P.; Chang, C. H.; Chu, S. I.
High-Speed Decoding of the Binary Golay Code
Centro de Ciencias Aplicadas y Desarrollo Tecnológico
Distrito Federal, México

Available in: http://www.redalyc.org/articulo.oa?id=47427855003
High-Speed Decoding of the Binary Golay Code

H. P. Lee¹, C. H. Chang¹, S. I. Chu²

¹ Department of Computer Science and Information Engineering, Fortune Institute of Technology, Kaohsiung 83160, Taiwan
² hpl@fotech.edu.tw
³ 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 445 bytes of memory. Therefore, one of the goals of this paper is to propose a novel message-syndrome decoding algorithm (MSDA) by using the message syndrome to correct the binary systematic Golay code.
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. Apparently, the decoding speed of the improved EDA 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

$$Q_{23} = \{ i | i \equiv \beta \mod 23 \text{ for } 1 \leq j \leq 22 \} = \{ 1, 2, 3, 4, 6, 8, 9, 12, 13, 16, 18 \}. \quad (1)$$

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

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

where $\beta$ 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 $0 \leq i \leq 22$, 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 $0 \leq i \leq 11$. $C(x)$ 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 $d(x)x^{12} + m(x) = (q(x)x^{15})g(x)$. 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 $12 \leq i \leq 22$.

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 $0 \leq i \leq 22$. The syndromes of the code are defined by

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

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 $m = (m_{11} \ldots m_1 m_0)$, $c = (c_{22} \ldots c_1 c_0)$, $e = (e_{22} \ldots e_1 e_0)$, $r = c + e = (r_{22} \ldots r_1 r_0)$, and $s = (s_{10} \ldots s_1 s_0)$, respectively. For the binary systematic Golay code, it follows from [2, p85] that the systematic generator matrix $G$ can be expressed as follows:
The mapping between the syndromes and error patterns under the error-correcting capability is 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 $1 \leq v \leq t - 1$ error patterns.

### 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 TDA, the following definition, theorems, and corollary given in [9] are needed.

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

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

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

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

$$w(a + b) = w(a) + w(b).$$ (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 $1 \leq v \leq t$. All $v$ errors occur in the $n - k$ parity-check bits if and only if the weight of syndrome $w(s) = v$.

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 $\sum_{i=1}^{3} \left(\binom{23}{i}\right) = 2,047$ 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 $(2047 \times (2 + 3)) = 10,235$ 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 $1 \leq v \leq t - 1$ error patterns.
Therefore, the LTDA needs \( \sum_{i=1}^{r} (2^3) = \sum_{i=1}^{2} (2^3) = 276 \) syndromes and their corresponding error patterns; that is, the FLT needs \( 276 \times (2 + 3) = 1,380 \) 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 than the LTDA. To overcome this problem, the MSDA is required 168 bytes. However, the decoding speed is reduced. Similarly, by Theorem 3, the latency of the cyclic codes to dramatically reduce the memory size of the RLT. By Theorem 3, see [2, p118], the memory size of the RLT, namely \( 10,235/23 = 445 \) bytes memory size [8]. Nonetheless, the memory size of the FLT is still large and can be further reduced. Therefore, the novel MSLT merely consists of 12 candidate syndromes which are the syndromes of having one error in the message part; that is, \( s_i \) is the syndrome of \( e_i \), which is one error located at bit 1 for \( 1 \leq i \leq 12 \). The MSLT is listed in Table 1.

### Table 1. MSLT for the Binary Systematic Golay Code.

<table>
<thead>
<tr>
<th>Syndrome</th>
<th>Syndrome</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s_1 = (01011100011) )</td>
<td>( s_2 = (10111000110) )</td>
</tr>
<tr>
<td>( s_3 = (00101101111) )</td>
<td>( s_4 = (01011001110) )</td>
</tr>
<tr>
<td>( s_5 = (10110111100) )</td>
<td>( s_6 = (00110011011) )</td>
</tr>
<tr>
<td>( s_7 = (01100110110) )</td>
<td>( s_8 = (11001110110) )</td>
</tr>
<tr>
<td>( s_9 = (11000111101) )</td>
<td>( s_{10} = (11010010101) )</td>
</tr>
<tr>
<td>( s_{11} = (11111001100) )</td>
<td>( s_{12} = (10101110001) )</td>
</tr>
</tbody>
</table>

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

**Lemma 2.** For the binary Golay code, let \( e \) be an error pattern, and let \( e_p \) and \( e_m \) be its parity-check part and message part, respectively. Let \( s_p \) and \( s_m \) be the syndromes corresponding to \( e_p \) and \( e_m \), respectively. Assume that \( w(e) \leq 3 \), then we have

- (i) \( w(s_p) = w(e_p) \);
- (ii) \( w(s_m) \geq d - w(e_m) \), where \( d = 7 \);
- (iii) \( (s_m << 12) + e_m \) 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 \((s_m << 12) + e_m)H^T = (s_m << 12) + e_m\) is always a codeword. Therefore, \( w(s_m) \geq d \), that is, \( w(s_m) \geq d - w(e_m) \). The proof is thus completed.
By Lemma 2, if \( n \leq t \), then one can obtain the following two properties.

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

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

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

Case 3: If \( w(s) > 3 \) and \( w(s_{di}) > 3 \), then 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 \( r^{(11)} \) and \( s^{(11)} \), respectively. Thus, one yields the following property.

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

Case 1: If \( w(s^{(11)}) = 2 \) 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 \( w(s^{(11)}) > 3 \) and \( w(s_{di}^{(11)}) \leq 2 \), then \( r \) has at least 2 errors in the message part.

Case 3: If \( w(s^{(11)}) > 3 \) or \( w(s_{di}^{(11)}) > 3 \), then at least 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 \( s = rh^{T} \) and its weight \( w(s) \).
3. If \( w(s) \leq 3 \), then return \( c = r - (s \ll 12) \) and go to step 13.
4. For \( 1 \leq i \leq 12 \), compute the syndrome \( s_{di} = s - s_{i} \) and its weight \( w(s_{di}) \).
5. If \( w(s_{di}) \leq 2 \), then return \( c = r - (s_{di} \ll 12) - (1 \ll (i - 1)) \), and go to step 13.
6. Cyclically shift \( r \) and \( s \) left by 11 bits obtaining \( r^{(11)} \) and \( s^{(11)} \), and then compute \( w(s^{(11)}) \).
7. If \( w(s^{(11)}) = 2 \) or \( 3 \), then return \( c = ((r^{(11)} - (s^{(11)} \ll 12)) \ll 12) - ((r^{(11)} - (s^{(11)} \ll 12)) >> 11) \) and go to step 13.
8. For \( 1 \leq i \leq 12 \), compute the syndrome \( s_{di}^{(11)} = s^{(11)} - s_{i} \) and its weight \( w(s_{di}^{(11)}) \).
9. If \( w(s_{di}^{(11)}) = 1 \) or \( 2 \), then return \( c = (r^{(11)} - (s_{di}^{(11)} \ll 12) - (1 \ll (i - 1)) \ll 12) - ((r^{(11)} - (s_{di}^{(11)} \ll 12) - (1 \ll (i - 1))) >> 11) \) and go to step 13.
10. Compute \( r' = r - 1 \) and \( s' = s - s_{1} \).
11. For \( 2 \leq i \leq 12 \), compute the syndrome \( s_{di}' = s' - s_{i} \) and its weight \( w(s_{di}') \).
12. If \( w(s_{di}') = 1 \), then return \( c = r' - (s_{di}' \ll 12) - (1 \ll (i - 1)) \).
13. Stop.

An example is given below. By (6), a message vector \( m = (1000000000000) \) is encoded into a binary systematic Golay code \( c = (10101110001100000000000000000000) \). Assume that a noise vector is \( e = (00010000000000001000000001) \), then the received word becomes \( r = c + e = (10111111001100010000000000000001) \). The decoding steps proceed as follows:

1. The received vector is \( r = (101111110011000100000001) \).
2. Compute the syndrome \( s = (10000001111) \) and its weight \( w(s) = 5 \).
3. Given that \( w(s) > 3 \), go to step 4.
4. For \( i = 1 \), compute the syndrome \( s_{di} = s - s_{1} = (11011101100) \) and its weight \( w(s_{di}) = 7 \).
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 $2^{12} \times \sum_{i=1}^{3} C_i^{23} = 8,384,512$ 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 $\mu$s per error pattern.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Number of errors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Proposed MSDA</td>
<td>0.3396</td>
</tr>
<tr>
<td>SDA with RSLT</td>
<td>0.7332</td>
</tr>
<tr>
<td>FLTDA with table</td>
<td>0.1774</td>
</tr>
<tr>
<td>SWDA with RLT</td>
<td>4.651</td>
</tr>
<tr>
<td>Improved EDA</td>
<td>16.32</td>
</tr>
</tbody>
</table>

Table 2. Computer Decoding Time for Decoding the Binary Golay Code (in $\mu$s).

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


