On the decoding of the (24,12,8) Golay code

Tsung-Ching Lin, Hsin-Chiu Chang, Hung-Peng Lee, Trieu-Kien Truong

Department of Information Engineering, I-Shou University, Kaohsiung County 840, Taiwan, ROC

Abstract

An improved syndrome shift-register decoding algorithm, called the syndrome-weight decoding algorithm, is proposed for decoding three possible errors and detecting four errors in the (24,12,8) Golay code. This method can also be extended to decode two other short codes, such as the (15,5,7) cyclic code and the (31,16,7) quadratic residue (QR) code. The proposed decoding algorithm makes use of the properties of cyclic codes, the weight of syndrome, and the syndrome decoder with a reduced-size lookup table (RSLT) in order to reduce the number of syndromes and their corresponding coset leaders. This approach results in a significant reduction in the memory requirement for the lookup table, thereby yielding a faster decoding algorithm. Simulation results show that the decoding speed of the proposed algorithm is approximately 3.6 times faster than that of the algebraic decoding algorithm.

1. Introduction

Linear cyclic block codes are practical for hardware or firmware implementations and therefore most widely used for the block codes. For the three-error-correcting cyclic codes, the binary (15,5,7) cyclic code, the (23,12,7) Golay code, and the (31,16,7) QR code are often used in communication systems. Among them, the binary (23,12,7) Golay code is a well-known cyclic code first discovered by Golay [6] in 1949. Such a 23-bit Golay code is a very useful code, particularly for a variety of applications in the past decades when a parity bit is added to each word to yield a half-rate code, called the (24,12,8) Golay code. One of the interesting applications is that it has been utilized to provide error control in the voyager mission, see [20].

Several decoding techniques have been developed to decode the Golay code [5,6,10,11,17,18]. It is well-known that all types of binary cyclic codes can be decoded by using Gröbner techniques up to their true minimum distance. In particular, the Gröbner technique for decoding the binary BCH codes is described in [12, pp. 69–91]. The key idea behind this technique is a systematic application of algorithmic procedures of Gröbner bases to obtain the error-location polynomial, denoted by \( L(z) \). The disadvantage of the Chen–Reed–Helleseth–Truong (CRHT) algorithm [2] is that it needs the process of transforming a set of syndrome polynomials \( F \) to the reduced Gröbner basis of the ideal \( \langle F \rangle \) generated by \( F \) with respect to the purely lexicographical order to automatically converges to \( L(z) \). It follows from [1] that Buchberger’s algorithm needed in the CRHT algorithm for generating Gröbner bases is the most computationally intensive. Therefore, it is very difficult to find the upper bound in the running time of Algorithm 2 of [2] used to generate Gröbner bases except for some special cases. To overcome this problem, one needs to develop a simpler algorithm for the special polynomial set \( F \). This is an important near-term problem of future studies. In 1987, Elia [5] developed a fast algebraic decoding algorithm to correct the binary (23,12,7) Golay code. Eventually, another decoding approach, called the shift-search decoding procedure [4,17], was developed to correct the...
binary QR code. As shown in [17], this shift-search procedure compares favorably in complexity and speed with Elia’s decoding method. The algebraic technique was shown to be slightly faster than the shift-search procedure. It is of interest to note that Elia’s decoding algorithm is simpler for the one- or two-error case, but it will become slower in decoding three errors because the computation complexity in $D^{1/3}$ increases rapidly. During the last two decades, some other algebraic decoding algorithms were proposed. For example, Lin et al. [7–9] and Reed et al. [2,3,14–18] used the Sylvester resultant method to solve nonlinear, multivariate equations to determine the error-locator polynomial. These algebraic decoding algorithms definitely required a large number of additions and multiplications over a finite field. It causes a time delay in decoding procedures and is rather complicated to implement in digital circuits. For some non-algebraic syndrome decoding methods, the syndrome decoder [20] and the Meggitt decoder [11] are two well-known syndrome decoders which apply in principle to any cyclic code. Basically, these two syndrome decoders result in minimal decoding delay and error probability. Obviously, one error detection and correction circuit is relatively simple to implement. However, the implementation complexity of the error pattern detection circuit and the size of the lookup table for the syndrome decoder will be increased rapidly when the length of codeword $n$ increases. The arithmetic decoding algorithm [20] is also a well-known decoder for decoding the $(24,12,8)$ Golay code, whereas, this arithmetic decoder has some disadvantages. First, it can only decode self-dual based codes which correct three errors or fewer errors. Second, it needs to compute a lot of matrix multiplications and additions for correcting up to three errors and detecting four errors, thereby increasing the decoding time.

In this paper, a new decoding algorithm called the syndrome-weight decoder is proposed to decode up to three possible errors in the $(15,5,7)$ cyclic code, the $(23,12,7)$ Golay code, and the $(31,16,7)$ QR code. The algorithm can be extended to detect four errors in the $(24,12,8)$ Golay code and the $(32,16,8)$ QR code. It is based on the properties of cyclic codes, the weight of syndrome, and the syndrome decoder with a RSLT. The refined lookup table (RLT) proposed in this paper consists of only 42 syndromes and their corresponding coset leaders, called representative error patterns, for decoding the $(23,12,7)$ Golay code. This leads to the reduction of memory size which renders this new decoding algorithm to be quite practical.

The remainder sections of this paper are organized as follows: The background of the algebraic decoding of the binary $(23,12,7)$ Golay code is given in Section 2. The syndrome decoder with a RSLT is briefly summarized in Section 3. Section 4 develops a syndrome-weight decoding algorithm for decoding the $(23,12,7)$ Golay code. Section 5 compares in software with existing decoding algorithms. Finally, a conclusion is presented in Section 6.

2. Algebraic decoding of the binary $(23,12,7)$ Golay code

For the binary $(23,12,7)$ Golay code of length 23 over $GF(2^{11})$, its quadratic residue set is the collection of all nonzero quadratic residues modulo $n$ given by

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

Let $\alpha$ be a primitive element in $GF(2^{11})$ such that $\alpha$ is a generator of the multiplicative group of $2^{11}-1$ nonzero elements in $GF(2^{11})$. A binary $(23,12,7)$ QR code is a cyclic code with the generator polynomial $g(x)$ of the form

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

where $\beta$ is a primitive 23rd root unity in the finite field $GF(2^m)$, with $m = 11$, the smallest positive integer such that $23|2^{11} - 1$.

The $(23,12,7)$ Golay code generated by $g(x)$ in this manner is a perfect code in a sense that the codewords and their three error correction spheres exhaust the vector space of 23-bit binary vectors. Because the minimum distance of the code is $d = 7$, the inequality $2\nu + 1 \leq 7$ is valid, where $\nu$ is the actual number of errors to be corrected. Hence, the $(23,12,7)$ Golay code allows for the correction of up to $t = [(d - 1)/2] = 3$, where $\lfloor x \rfloor$ denotes the greatest integer less than or equal to $x$.

Let the polynomial $c(x) = c_0 + c_1 x + \cdots + c_{23} x^{23}$ be a codeword which is a multiple of the generator polynomial $g(x)$, where $c \in GF(2)$. Also, let the polynomials $i(x) = i_0 + i_1 x + \cdots + i_{11} x^{11}$ and $p(x) = p_0 + p_1 x + \cdots + p_{10} x^{10}$ be the information and parity-check polynomials of a codeword $c(x)$, respectively, where $i, p \in GF(2)$. Then the codeword $c(x)$ can also be represented as $c(x) = i(x)x^{11} + p(x)$, where $p(x)$ is a remainder polynomial of degree less than 11 obtained by dividing $i(x)x^{11}$ by $g(x)$.

The Elia decoding algorithm was utilized to decode the Golay code with no more than three errors. To illustrate this method, we define $e(x) = \sum_{i=0}^{22} e_i x^i$ to be the error polynomial. The received word then has the form $r(x) = c(x) + e(x)$, where $r(x) = \sum_{i=0}^{22} r_i x^i$.

The set of known syndromes obtained by evaluating $r(x)$ at the roots of $g(x)$ is given by

$$S_i = r(x^i) = c(x^i) + e(x^i) = e(x^i), i \in Q_{23}. \quad (3)$$

Assume that $\nu$ errors occur in the received word. The plain error-locator polynomial is defined to be a polynomial of degree $\nu$; that is

$$L(z) = \prod_{i=1}^{\nu} (z - z_i) = z^\nu + \sum_{j=1}^{\nu} c_j z^{\nu-j}, \quad (4a)$$
where \(1 \leq i \leq v\) and \(Z_i\) are the locations of the \(v\) errors. Note that the general error-locator polynomial of the cyclic codes found in [13] is given by

\[
L(z) = \prod_{i=1}^{n} (z - Z_i^{-1}).
\]  

(4b)

Given the known syndromes \(S_1, S_3, \) and \(S_8\), Elia’s algorithm for decoding the (23, 12, 7) Golay code [5] is summarized as below.

\[
L(z) = \begin{cases} 
0, & \text{no errors if } S_1 = 0, \\
z + S_1, & \text{one error if } S_1^3 = S_3 \text{ and } S_1^0 = S_9, \\
z^2 + S_1z + (S_3^2 + S_5^2), & \text{two errors if } S_1D^{1/3} = S_3, \\
z^3 + S_1z^2 + (S_3^1 + D^{1/3})z + (S_3 + S_1D^{1/3}): & \text{otherwise. three errors.}
\end{cases}
\]  

(5)

Here, \(D = S_1^3 + S_3^1 + \left( S_1^0 + S_9 \right) / \left( S_3^1 + S_9 \right) \in GF(2^{11})\) and the cube root \(D^{1/3}\) in \(GF(2^{11})\) is equal to \(D^{1365}\) which may be implemented directly. However, the exponent \(D\), namely 1365, is so large that the multiplicative complexity of the direct method is very high. Towards this end, Fermat’s theorem given in [19] can be used to compute the cube root \(D^{1/3}\) needed in Elia’s decoding algorithm.

A (24, 12, 8) Golay code can be formed by adding an overall parity-check bit to the (23, 12, 7) Golay code. The following theorem 1. Syndrome decoding with a reduce-size lookup table

Theorem 1. Let \(s(x)\) be the syndrome polynomial corresponding to a received polynomial \(r(x)\). Also, let \(r^{i/2}(x)\) be the polynomial obtained by cyclically shifting the coefficients of \(r(x)\) one bit to the right. Then the remainder obtained when dividing \(xs(x)\) by \(g(x)\) is the syndrome \(s(x)\) (x) corresponding to \(r^{i/2}(x)\).

The RSLT generated by a C++ program for decoding the (23, 12, 7) Golay code is shown in Table 1 as given in the Appendix. For simplicity, let the message, codeword, transmitted error, and received word be expressed as the binary vector forms \(m = (m_0, m_1, \ldots, m_k)\), \(c = (c_0, c_1, \ldots, c_n)\), \(e = (e_0, e_1, \ldots, e_n)\) and \(r = (r_0, r_1, \ldots, r_n)\), respectively. The codeword \(c\) is generated by a systematic generator matrix \(G\) and the message is at the bottom \(k\) bits of \(c\), i.e., \((c_{n-k}, c_{n-k+1}, \ldots, c_{n-1})\). The syndrome decoding algorithm associated with the RSLT for decoding the (23, 12, 7) Golay code is summarized as follows:

1. Initially set counter to be \(i = 0\) and give a received word \(r\).
2. Compute the syndrome of \(r^{i/2}\); that is, \(s^{i/2} = r^{i}(\beta)\).
3. If \(s^{i/2} = 0\), no errors occur and then go to step 7.
4. Look up Table 1. If \(s^{i/2}\) is found in the table, then look up \(e_i\) from the same table and go to step 6.
5. Increment the counter \(i\) by one. Cyclically shift the syndrome left by one bit and then calculate the residue of this result mod \(g(x)\), Go to step 4.
6. Cyclically shift \(e_i\) left by \(i\) bits to obtain the corrected error pattern \(e\) and then subtract \(e\) from \(r\) to obtain the correct codeword \(c\).
7. Stop.
4. Syndrome-weight decoding algorithm

This proposed algorithm, called the syndrome-weight decoding algorithm, utilizes the properties of the cyclic code, the weight of syndrome, and the syndrome decoder with a RSLT to reduce the number of the syndromes and their corresponding coset leaders in the RSLT. It can correct up to three possible errors and detect four errors in the binary (24,12,8) Golay code and the (32,16,8) QR code, and then it can also correct up to three possible errors in the binary (15,5,7) cyclic code. As shown in this section, the novel lookup table, called the refined lookup table (RLT) used to implement the proposed syndrome-weight decoder, can be obtained from the RSLT of the syndrome decoder.

Upon inspection of Table 1, it is obvious that if the weight of coset leader is \(w(e_i) = 3\) for \(13 \leq i \leq 89\), then one can delete the coset leaders that have one error in the \(n - k\) parity check bits. Thus, the error positions in the \(k\) message bits of these deleted error patterns can be substituted by the weight of the coset leader \(w(e_i) = 2\) for \(2 \leq j \leq 12\). For example, from Table 1, one observes that \(s_{33} - s_{2} = (00000000001)\) and its weight \(w(s_{33} - s_{2}) = 1\) which implies that only one error occurs in the \(n - k\) parity check bits of \(e\). In this case, one can replace this coset leader \(e_{2}\) by \(e_{2}\) while decoding, and then \(e_{2}\) can be omitted. Similarly, the coset leaders from \(e_{34}\) to \(e_{2}\) can be also replaced by \(e_{2}\). The procedure described in Table 1 is repeated recursively, the coset leaders from \(e_{24}\) to \(e_{9}\) can be replaced by \(e_{7}\), and \(e_{6}\), respectively. Therefore, the RLT for decoding the binary (23,12,7) Golay code, as given in Table 2 of the Appendix, only consists of 42 syndromes corresponding to coset leaders. The size of the RSLT for the binary (15,5,7) cyclic code and (31,16,7) QR codes are thus \(R_{15}/15 = \sum_{i=1}^{15} \left( \begin{array}{c} 15 \\ i \end{array} \right) /15 = 39\) and \(R_{31}/31 = \sum_{i=1}^{31} \left( \begin{array}{c} 31 \\ i \end{array} \right) /31 = 161\), respectively.
obtained by subtracting this received word left by one bit. Increased next syndrome, namely 

However, the size of the refined lookup tables for the binary (15,5,7) cyclic code and the binary (31,16,7) QR code are 9 and 72, respectively. By using a message length \( k \), one can obtain an equation for the size of the RLT for these codes given by

\[
N = \left(k - \left\lfloor \frac{k}{2} \right\rfloor \right) \left(1 + \left\lfloor \frac{k}{2} \right\rfloor \right) = \left\lceil \frac{(k + 1)^2}{4} \right\rceil. \tag{6}
\]

Table 3 shows the comparison of the size between the RSLT and the RLT in three different codes.

The syndrome-weight decoding algorithm proposed in this paper is described as follows:

Given a received word \( r \), initially set counter to be \( i = 0 \). First, the syndrome is computed directly and then the weight of this syndrome \( w(s) \) is calculated. If \( w(s) = 0 \), no error is occurred in the received word. If \( w(s) \leq 3 \), then \( 3 = (7 - 1)/2 \) is the error capability of this code, then this tactic implies that at most 3 errors occur in the parity check block of the received word. In this case, the syndrome is shifted left by \( k \) bits to form a \( n \)-bit length word, and the corrected codeword is then obtained by subtracting this \( n \)-bit length word from the received word. The syndrome difference, denoted by \( s_{ij} \) for \( 1 \leq i \leq 42 \), is then computed by subtracting this syndrome from the first syndrome patterns, denoted by \( s_i \), in the RLT. Finally, the difference of this new weight of this syndrome, denoted by \( w(s_{ij}) = w(s \oplus s_i) \), where \( \oplus \) denotes the EXOR operator, is computed. Three cases are needed to be considered as follows:

Case 1: If the \( k \) message bits occur one error, then the remaining \( n - k \) parity check bits probably have 0, 1, or 2 errors; that is, \( w(s_{ij}) \leq 2 \).

Case 2: If the \( k \) message bits occur two errors, then the remaining \( n - k \) parity check bits probably have 0 or 1 error; that is, \( w(s_{ij}) \leq 1 \).

Case 3: If the \( k \) message bits occur three errors, then the remaining \( n - k \) parity check bits probably have no errors; that is, \( w(s_{ij}) = 0 \).

If one of the above mentioned three cases is satisfied, then one corrects the received word; otherwise the difference of the next syndrome, namely \( w(s_{ij}) \), is computed and is repeated to check the three cases again. If no any cases are satisfied after checking the final \( w(s_{ij}) \), then one shifts the syndrome left by one bit which means that one proceeds a cyclic shift of the received word left by one bit. Increased \( i \) by one, the previous procedure is repeated until the received word is corrected.

\[
\begin{array}{c|c|c|c}
\text{Codes} & \text{RSLT} & \text{RLT} & \text{Decreased percentage (\%) } \\
\hline
(15,5,7) & 39 & 9 & 76.9 \\
(23,12,7) & 300 & 42 & 52.8 \\
(31,16,7) & 161 & 72 & 55.3 \\
\end{array}
\]
The proposed syndrome-weight decoding algorithm for decoding the (23,12,7) Golay code, the (15,5,7) cyclic code, and the (31,16,7) QR code is stated explicitly as follows:

1. Initially set counter to be $i = 0$ for $0 \leq i \leq n - 1$ and give a received word $r$.
2. Compute the syndrome of $r^{(i)}$; that is, $s^{(i)} = r^{(i)}(\beta)$.
3. If $w(s^{(i)}) = 0$, no errors occur and then go to step 9.
4. If $w(s^{(i)}) \leq 3$, there are errors occurred in the parity check section, then $c = r \oplus (\text{cyclicly shift } s^{(i)} \text{ left by } (k - i) \text{ bits})$ and go to step 9, else set $j = 0$.

![Flowchart](image)

Fig. 1. Flowchart of the proposed syndrome-weight decoding algorithm with the RLT.
Compute the syndrome difference $s_{dj} = s^{(i)} \oplus s_j$ for $1 \leq j \leq N$, thereby yielding $w(s_{dj})$.

If $w(s_{dj}) \leq 2$, then the corrected codeword equals to $r \oplus (e_j \gg i) \oplus (s_{dj} \ll (k - i))$ and go to step 9.

Increment the counter $j$ by 1 and go to step 5.

Increment the counter $i$ by 1. Cyclically shift the syndrome left by one bit and then calculate the residue of this result $\mod g(x)$. Compute the new syndrome and the weight of this new syndrome and go to step 4.

Stop.

The flowchart of the proposed syndrome-weight decoding algorithm with the RLT is depicted in Fig. 1.

### 5. Simulation results

The proposed decoding algorithm written in C++ language is implemented on an Intel E6750 PC. All codes with errors were created for decoding the (23,12,7) Golay code, the (15,5,7) cyclic code, and the (31,16,7) QR code perfectly with an averaged speed of 6.51 $\mu$s, 1.12 $\mu$s, and 18.73 $\mu$s per error, respectively. These results are shown in detail in Table 4 and are in compared also with the syndrome decoding algorithm and Elia’s algorithm.

### 6. Conclusions

A syndrome-weight decoding algorithm was developed to decode up to three possible errors in the (15,5,7) cyclic code, (23,12,7) Golay code, and (31,16,7) QR code. Also, it can detect four errors in the (24,12,8) Golay code and (32,16,8) QR code. It is shown in this paper that using the proposed algorithm, the size of the RSLT needed in the syndrome decoder can be further reduced by 52%. In addition, the difference in weight of syndrome is also utilized to obtain the possible error pattern needed in each search process. Furthermore, the proposed method is compared in software to Elia’s decoding algorithm with Fermat’s theorem and the syndrome decoding algorithm with a RSLT. Computer simulations show that the speed of the proposed algorithm is much faster than Elia’s algorithm and slightly slower than the syndrome decoding algorithm with a RSLT. Among those three decoding algorithms, Elia’s decoding algorithm does not need any lookup table, but it requires more computational time to calculate the decoding procedure even if one uses Fermat’s theorem to compute the cube root $D^{1/3}$ in $GF(2^{11})$. Although the syndrome decoding algorithm with a RSLT surpasses the other two algorithms in decoding speed, it still needs a moderate memory requirement for the RSLT. The proposed syndrome-weight decoding algorithm is an improved syndrome shift-register decoding algorithm and gives an extremely fast decoding algorithm that might apply to decode all small cyclic codes including QR codes and BCH codes. It is expected that this proposed decoding algorithm can be extended probably to other large cyclic codes in the future. However, the memory size of the RLT will become bigger that significantly increases the decoding complexity in hardware. Therefore, it is worthy of much more investigation on whether there exists an efficient way for generalizing the proposed algorithm. From an engineering point-of-view, both the time and space complexities of this proposed decoder must be evaluated.

### Acknowledgements

The authors thank the Editor-in-Chief, Dr. W. Pedrycz, and the anonymous reviewers for the useful comments and suggestions that improved the quality as well as the presentation of this paper.

### Appendix A

Tables 1 and 2.

### References

