Using Hash Table and Cyclotomic Coset method for decoding the quadratic residue code

Yan-Haw Chen, C. D. Lee, J. J. Wang and Z. W. Kang
Dept. of Information of engineering, I-Shou university Kaohsiung 84008, Taiwan

ABSTRACT

A method for decoding of the quadratic residue (QR) code with hash tables is presented. The method can be applied in decoding the (31, 16, 7) QR code that the generator polynomial can be factored. In other words, the mapping between elements of syndrome S_1 and all correctable error patterns is not one-to-one. Therefore, the decoding of the (31, 16, 7) QR code needs to stuck S_1, S_5, S_7 known syndrome mapping an error pattern that has one-to-one nature, where a subscript 1, 5, 7 are cyclotomic cosets. Furthermore, the algorithm determines the error locations by hash tables without operating the additions and multiplications over finite field. To decode QR code result, the hash table method for the (31, 16, 7) QR code is dramatically reduced the memory size above 97%. It is very suitable for high speed in modern communication system

INTRODUCTION

Recently, an algebraic decoding of QR code has been proposed by [1] to decode the some binary QR codes which use irreducible generator polynomials. The paper in [1] is computing the unknown syndromes with an efficient Berlekamp–Massey algorithm [2–4] to obtain the error-locator polynomial. But, these decoding methods are depended on matrix operation; as a result, it would be difficult to implement in embedded system. As previously described by Chen *et al*. [5–6], using a one-to-one mapping between the error patterns and syndromes, the lookup table decoding (LTD) with inject error method significantly reduces the memory size requirement in processor of the embedded system and is thus particularly suitable for the development in embedded system written in C language. Binary search is an efficient search method, which has the advantage of increasing performance and plays an important role in finding the error patterns directly and precisely for QR codes. A newly proposed hash table (HT) method is introduced for decoding the four binary systematic QR codes with parameters (23, 12, 7), (41, 21, 9), (47, 24, 11), and (71, 36, 11) are only used by generator polynomial $g(x)$ that is also said to be irreducible polynomial [7]. In this paper, the proposed method can be used for decoding binary (31, 16, 7) QR codes with factoring generator polynomials. The simulation result shows that the hash search would be better than a binary search for the decoding of QR codes.

HASH TABLE FOR DECODING

The encoding of the QR code utilized systematic, in which $m(x)=m_0+m_1x,..,m_{k-1}x^{k-1}$ is the information message polynomial. The systematic encoding can easily obtain the code polynomial $c(x)=x^{k-1}m(x)-d(x)$ by $d(x)\equiv m(x)x^{k-1} \bmod g(x)$. The polynomial $d(x)$ is said the remainder polynomial. If $c(x)$ is transmitted through a noisy channel and the $r(x)=c(x)+e(x)=r_0+r_1x+\cdots+r_k^kx^k+\cdots+r_{n-1}x^{n-1}$ and the error polynomial is $\mathbf{e}(x)=(\mathbf{e}_d(x),\mathbf{e}_m(x))$, where $e_d(x)=e_0+e_1x+\cdots+e_{k-2}x^{k-2}$, $e_m=e_m(x)=e_{k-1}+e_kx^k+\cdots+e_{n-12}x^{n-1}$.

A. To make table of the value of the beta for decoding

The Table 1 shows the value of beta for decoding QR code. The (31, 16, 7) QR code needs β^i, $\beta^{(5i \bmod 31)}$ *and* $\beta^{(7i \bmod 31)}$ for computing the value of syndrome, where $0 \le i \le 30$. The value of β^i is presentation Hexadecimal.

TABLE 1. VALUES OF β *TO* THE i-TH POWER FOR THE QR CODE

β^0 =01	β^8 =0D	β^{16} =1B	β^{24} =1E
β^1 =02	β^9 =1A	β^{17} =13	β^{25} =19
β^2 =04	β^{10} =11	β^{18} =03	β^{26} =17
β^3 =08	β^{11} =07	β^{19} =06	β^{27} =0B
β^4 =10	β^{12} =0E	β^{20} =0C	β^{28} =16
β^5 =05	β^{13} =1C	β^{21} =18	β^{29} =09
β^6 =0A	β^{14} =1D	β^{22} =15	β^{30} =12
β^7 =14	β^{15} =1F	β^{23} =0F	

B. Making hash table

The value of the key is evaluated by a received polynomial $r(x)$ as follows:

$$key = S = (r(\beta^7), r(\beta^5), r(\beta)) = (\sum_{k=1}^{t-1}\beta^{7u_k}, \sum_{k=1}^{t-1}\beta^{5u_k}, \sum_{k=1}^{t-1}\beta^{u_k}) \quad \text{for } 0 \le u_1 < u_2 < \cdots < u_{t-1} < n \tag{1}$$

where u_k stands for error location. The proposed decoding method is based on a one-to-one mapping between the *key* and error patterns. The proposed hash function can be utilized in the decoding (31, 16, 7) QR code as follows:

$$f(key) \equiv key \bmod 2^9 \tag{2}$$

According to (2), the hash table can be made by modulo 512 that is only collisions 1 times. A node of the hash table can be designed as Table 2. Using equation (2) to construct the hash table is called the array HT[] that can be for finding error in data communication.

TABLE 2. AN ENTRY FORMAT FOR HASH TABLE

Syndrome Q bits	Error patterns P bits	Next index I bits
6 bits	8 bits	7 bits

The value of P	Error patterns
0	1000000000000000
1	0100000000000000
⋮	⋮
15	0000000000000001
⋮	⋮
136	0000000000000011

Fig. 1. Using the value of P as index to lookup error pattern.

USING HASH TABLE DECODING THE QR CODE

C. *the decoding of the QR code*

The largest value of syndrome $S^{(v)}$, for $v = 1,2,3$, as Q bits in formatting entry which needs more memory to store. Therefore, the largest value of syndrome divided by 512 has Q bits in formatting an entry. However, all original syndromes S needs to recover by $S = Q * 512 + f(key)$. The decoding of the QR code algorithm as follows:

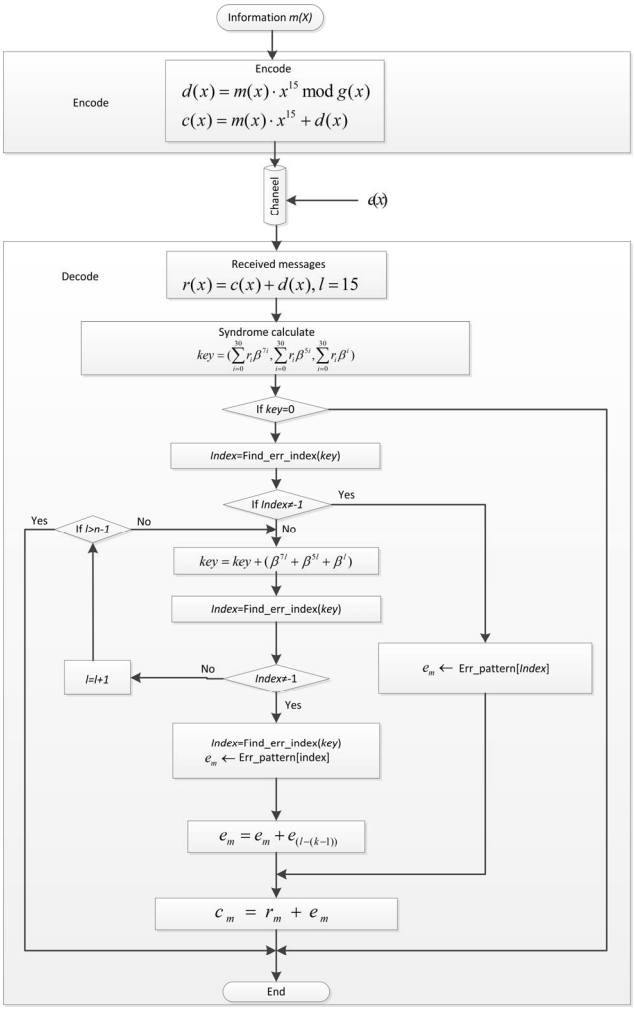

Fig. 2. Decoding of the QR code method

Find_err_index(*key*) procedure as listed below:

Step1. Set $i = 0$, *target=f*(*key*), *Next_ target* = *target*.
Step2. If next index *I* of the HT[*Next_target*] = -1, *Next_ target* = *P*, and proceed to Step 7.
Step3. *S* = *Q**512 + *target*, the value of *Q* is obtained from the HT[*Next_target*].
Step4. If *key* = *S*, then the value of *P* is as index for finding error pattern from HT[*Next_target*], *Next_ target* = *P*, and proceed to Step 7.
Step5. If the index *I* portion of HT[*Next_target*] = -1, then *Next_target* = -1 and proceed to Step 7.
Step6. *Next_target* ⟵ *Next_target* + 512, *i* = *i* + 1, proceed to Step 3.
Step7. Return *Next_ target*.

ANALYSIS AND DISCUSSION OF EXPERIMENTAL RESULT

The computer simulations among different decoding methods for the (31, 16, 7) QR code are compared in terms of the decoding time and memory size as shown in Table 3. In the intel i5 experiment of simulation, one tests 1,000,000 codeword for each error case. It is demonstrated that this new decoding scheme is suitable for both software and hardware realizations. Table 3 gives a comparison between the direct method and the binary search method for every number of error cases to be found. Based on the experiments that tested a million codewords, a comparison with various different number errors is given in Table 3. Although, in the worst case, the hash search algorithm is about 15% slower than in decoding time for the direct method, the memory size is reduced approximately 97% of the one when compared with the direct search. The hash search algorithm for decoding times of the QR code is approximately 55% faster than in using the binary search method, the memory size requirement is reduced approximately 24% of the one comparable to the binary search. The memory sizes in the above methods require Direct search 2^{15}*2 Bytes, Hash search 496*2+120*2+137*2 Bytes, Binary search 496*2+496*2 Bytes, respectively.

TABLE 3. COMPUTER'S TIME REQUIRED TO DECODE (31, 16, 7) QR CODE.

Decoding method	Decoding time (s)	Memory size (byte)
Direct search	0.45	64 Kbytes
Hash search	0.53	1.47 Kbytes
Binary search	1.19	1.94 Kbytes

ACKNOWLEDGMENT

This paper is supported in part by Taiwan's Ministry of Science and Technology MOST-106-2221-E-214-004.

REFERENCE

[1] Y. H. Chen, T. K. Truong, Y. Chang, C. D. Lee, S. H. Chen, "Algebraic decoding of quadratic residue codes using Berlekamp–Massey algorithm," *J. Inf. Sci. Eng*., vol. 23, 2007, pp. 127–145.

[2] E. R. Berlekamp, *Algebraic decoding theory*. McGraw-Hill, New York, 1968, 1st edn.

[3] J. L. Massey, "Shift-register synthesis and BCH decoding," *IEEE Trans. Inf. Theory*, vol. 15, 1969, pp. 122–127.

[4] X. Youzhi, "Implementation of Berlekamp–Massey algorithm without inversion," *Proc. IEE Commun. Speech Vis*., vol. 138, 1991, pp. 138–140.

[5] Y. H. Chen, T. K. Truong, C. H. Huang, "A lookup table decoding of systematic (47, 24, 11) quadratic residue code," *Inf. Sci.*, vol. 179, 2009, pp. 2470–2477.

[6] Y, H. Chen, C. H. Chien, C. H. Huang, "Efficient decoding of systematic (23, 12, 7) and (41, 21, 9) quadratic residue codes," *J. Inf. Sci. Eng*., vol. 26, 2010, pp. 1831–1843.

[7] Y. H. Chen, T. K. Truong, "Fast algorithm for decoding of systematic quadratic residue codes," *IET Commun*., vol. 5, 2011, pp. 1361–1367