http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 9 (6): 1251-1254, 2010
ISSN 1812-5638

Improved Decoding Algorithm for BCH Code

[1]Du Yatao, [1]Zhong Zhi, [1]Zhang Hailong, [1]Gong Fang, [2]Ren Guanghui and [3]Lan Shulei
[1]Information and Communication Engineering College, Harbin Engineering University, Harbin, China
[2]School of Electronical and Information Technology, Harbin Institute of Technology, Harbin, China
[3]Department of Math, Arts and Science College, Harbin Normal University, Harbin, China

Abstract: To improve the decoding speed and reduce the memory cost of the look-up table decoding method for binary BCH code, an improved decoding algorithm which can correct three possible errors in (15,5,7) BCH code is proposed. The algorithm makes use of the properties of system codes, the hamming weight of syndrome patterns and the look-up table to decode. The reduced look-up table contains syndrome patterns and corresponding error patterns which have errors occurred only in the message block of the received codeword, which makes the proposed algorithm reduce about 95% memory size compared with the traditional look-up table algorithm. Moreover, the hardware decoder based on this algorithm has a low complexity and a great fast decoding speed. The FPGA simulation shows that the decoding rate is 375 MHz at the 50 MHz system clock rate.

Key words: Look-up table, memory cost, error pattern, syndrome pattern, hamming weight

INTRODUCTION

The BCH codes are a class of error-correcting codes possessing strict algebraic architecture and are easy to construct (Peterson, 1960). The BCH codes are widely used in data transmission due to its powerful correction abilities especially the performance of short codes closing to theory values (Seddiki *et al*., 2008).

In all the iterative decoding algorithms for binary BCH code, the Berlekamp-Masssey iterative algorithm and Chien's search algorithm are the most efficient (Blake *et al*., 1998; Penzhorn, 1993). However, both of them need complex algebraic operation and iterative operation. Another algebraic decoding method, known as the step-by-step decoding method was presented and improved for the general cases of BCH codes (Chr *et al*., 2004; Xiaobei *et al*., 2007). The method is less complex since it avoids calculating the coefficients of error location polynomial and searching the roots. However the decoding speed is not fast enough. Then the arithmetic algorithm based on look-up table (Hung *et al*., 1999) was proposed, which has fast decoding speed and is simple in structure and easy in implementation. The drawback of this algorithm is that it requires a great deal of EMS memory.

To avoid this problem, A weight method (Ozkan and Ozkan, 2002) using a reduced look-up table to decode the three possible errors in (15,5,7) BCH code is proposed by Lee *et al*. (2008) and Chang *et al*. (2008). The data in his reduced look-up table consists of syndrome patterns and corresponding error patterns which only have one and two errors occurred in the message block of the received codeword. It can greatly reduce the memory size, but it is restricted to the situation that one error occurred in the message block and two errors occurred in the parity check block. The algorithm proposed in this study needs so many circulations that the computing time is increasing enormously. To resolve the issue a modified decoding algorithm also based on look-up table and hamming weight is developed in this study. In the algorithm the look-up table consists of syndrome patterns and corresponding error patterns of the codeword which has one to three errors occurred in the message block. Compared with Lee's table, our method increases only a few memory cells but the decoding speed can be significantly improved. The decoding algorithm just needs some addition (modulo 2) with the look-up table and computation of the weight of the word. Therefore, the decoder based on this algorithm is simpler, taking fewer memory cells and suitable for hardware implementation.

BACKGROUND OF ENCODING AND DECODING FOR BINARY BCH CODE

Coding theory of binary BCH code: For any integer $m \geq 3$ and $t < 2^m-1$ it exists a primitive BCH code with the parameters: $n = 2^m-1$, $n-k \leq 2t+1$, $d_{min} \geq 2t+1$. This code can correct t or fewer errors over a span of 2^m-1 bit positions.

Corresponding Author: Lan Shulei, Department of Math, Arts and Science College, Harbin Normal University, Harbin, China
Tel: 86-451-82589812

Supposing the generator polynomial is g(x), each of the code polynomial c(x) is a multiple of g(x) and can be expressed as c(x) = m(x) g(x), where, m(x) is the message polynomial whose coefficients form the message vector m. If the generator matrix is G, the codeword can also be achieved by c = m×G.

Suppose a codeword c is transmitted over a noise channel, the received codeword is then r = c+e, where, e is the error pattern induced by the noisy channel. The decoding procedure mainly includes three parts: (1) Syndrome computation, (2) detemination of error pattern and (3) error correction.

Decoding algorithm with traditional look-up table: Take binary (15,5,7) BCH code as the example, the system generator matrix of (15,5,7) BCH code is:

$$G=\begin{bmatrix}1&0&0&0&0&1&1&1&0&1&1&0&0&1&0\\0&1&0&0&0&0&1&1&1&0&1&1&0&0&1\\0&0&1&0&0&1&1&0&1&0&1&1&1&1&0\\0&0&0&1&0&0&1&1&0&1&0&1&1&1&1\\0&0&0&0&1&1&1&0&1&1&0&0&1&0&1\end{bmatrix}$$

And the parity check matrix is:

$$H=\begin{bmatrix}1&0&1&0&1&1&0&0&0&0&0&0&0&0&0\\1&1&1&1&1&0&1&0&0&0&0&0&0&0&0\\1&1&0&1&0&0&0&1&0&0&0&0&0&0&0\\0&1&1&0&1&0&0&0&1&0&0&0&0&0&0\\1&0&0&1&1&0&0&0&0&1&0&0&0&0&0\\1&1&1&0&0&0&0&0&0&0&1&0&0&0&0\\0&1&1&1&0&0&0&0&0&0&0&1&0&0&0\\0&0&1&1&1&0&0&0&0&0&0&0&1&0&0\\1&0&1&1&0&0&0&0&0&0&0&0&0&1&0\\0&1&0&1&1&0&0&0&0&0&0&0&0&0&1\end{bmatrix}$$

From the matrix above, we can see the generator matrix version is G = [I_k P], so the first five bits of the generated codeword is the message block and the remaining is the parity check block. If the codeword c = (c_1 c_2 .... c_{15}) is transmitted, then the received codeword is r = c+e, where, is the error pattern.

The calculating formula of the syndrome is $S = r\times H^T = e\times H^T$. From the calculating formula, we can see that if S = 0, then e_1 e_2 .... e_{15} are all equal to zero, it means there is no error happened in the received codeword. And if S ≠ 0, then e_1 e_2 .... e_{15} are not all equal to zero, it means there is at least one error happened in the received codeword. The relationship between the syndrome and the parity check matrix is that if there is one error in the location i of the codeword, the S will be the same as the i'th column in H and if there are two errors in the location i and j of the codeword, the S will be the summation (modulo 2) of the i'th and j'th column in H and similar rules apply to the situation when there are three errors. Then all the syndrome patterns and corresponding error patterns are stored in the look-up table. The decoding process is to compute S first and look for the same data in the look-up table one by one, finally use the selected error pattern to correct errors. For (15,5,7) BCH code, there are all 575 syndrome patterns and corresponding error patterns stored in the traditional look-up table.

IMPROVED DECODING ALGORITHM

The look-up table used in this improved algorithm consists of syndrome patterns and corresponding error patterns which have one to three errors occurred in the message block of the codeword. Then the look-up table contains only 25 syndrome patterns and corresponding error patterns. Suppose that there are only three or less errors occurred in (15,5,7) BCH codeword. Due to the latter part of H is a 10×10 identity matrix and $S = eH^T$, if the weight of S w(S)≤3, it means at most three errors only occurred in the parity check block and the location of 1 in S is just the error location in the parity check block. Then shift the syndrome right 5 bits to form a 15-bit length word and minus (modulo 2) the received codeword to decode. If w(S)≥4, it means at least one error occurred in the message block. First, the syndrome minus (modulo 2) all syndrome patterns in the table to obtain the difference and compute the weight of these difference, respectively. Then we will discuss the errors location in three possible conditions: (1) One error in the message block and two or less errors in the check block, the weight of the corresponding difference will be the number of errors occurred in the parity check block. (2) Two errors in the message block and one or no error in the check block, the weight of the corresponding difference will be one or zero. (3) Three errors in the message block and no error in the check block, the weight of the corresponding difference will be zero. Also, the location of 1 in the difference is just the error location in the parity check block. At last shift the certain difference right 5 bit to form a 15-bit length word, then minus (modulo 2) the received codeword and minus (modulo 2) the corresponding error pattern to decode.

The proposed decoding algorithm for (15,5,7) BCH code is presented as follows:

Step 1: Receive a codeword r
Step 2: Compute $S = rH^T$ and w(S)
Step 3: If w(S) = 0, then go to end
Step 4: If w(S)≤3, compute c = r-1(S>>5), then go to end
Step 5: Compute all $S_d = S-S_i$ and all $w(S_d)$, respectively
Step 6: Choose the certain S_d' and corresponding e'
Step 7: Compute c = r-(S_d'>>5)-e'
Step 8: End

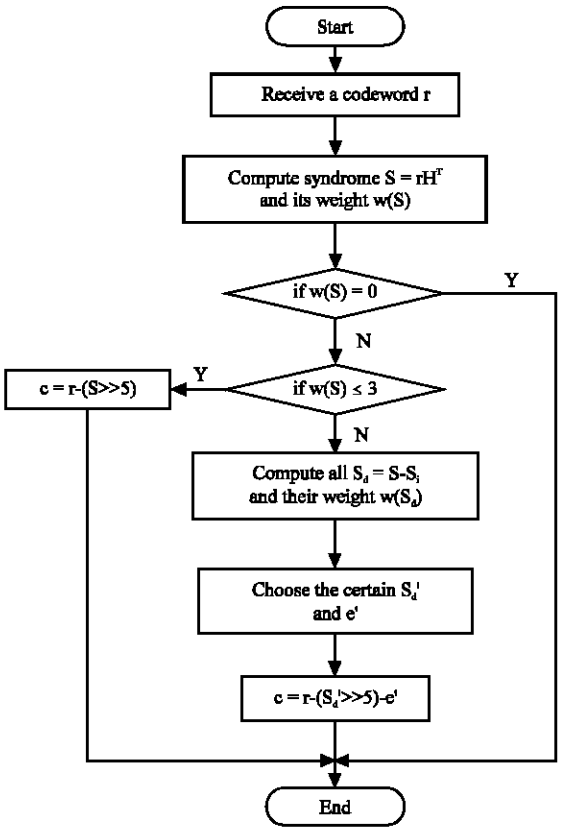

Fig. 1: The flowchart of the algorithm

The flowchart of this algorithm is shown in Fig. 1.

Give an example: There is a simple example to expatiate the whole decoding process.

A message m = (0 0 0 0 1) is encoded into a codeword c = (0 0 0 0 1 1 1 0 1 1 0 0 1 0 1) by an encoder. If the received codeword is r = (1 0 0 0 1 0 1 0 1 1 1 0 1 0 1), then the decoding procedure is:

Step 1: Receive the codeword r = (1 0 0 0 1 0 1 0 1 1 1 0 1 0 1)

Step 2: Compute S = (0 1 1 0 1 0 0 0 1 0) and w(S) = 4

Step 3: w(S)≠0, go to step 4

Step 4: w(S)>3, go to step 5

Step 5: Compute all S_d and their weight $w(S_d)$

$$S_{d1} = (1\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0),\ w(S_{d1}) = 2$$

$$S_{d2} = (0\ 0\ 0\ 1\ 1\ 1\ 1\ 0\ 1\ 1),\ w(S_{d2}) = 6$$

$$\vdots$$

$$S_{d25} = (0\ 0\ 0\ 0\ 1\ 1\ 0\ 1\ 1\ 0),\ w(S_{d25}) = 4$$

Step 6: $w(S_{d1}) = 2=3$, Choose the certain $S_d' = S_{d1}$ =(1 0 0 0 0 1 0 0 0 0) and the corresponding e' = (1 0 0 0 0 0 0 0 0 0 0 0 0 0 0)

Step 7: Compute c = (1 0 0 0 1 0 1 0 1 1 1 0 1 0 1) - ((1 0 0 0 0 1 0 0 0 0)>>5) - (1 0 0 0 0 0 0 0 0 0 0 0 0 0 0) = (0 0 0 0 1 1 1 0 1 1 0 0 1 0 1) then go to end

RESULTS AND DISCUSSION

The improved look-up table algorithm for (15,5,7) BCH codes can directly decode a received codeword (with three or less errors) without knowing the number of errors or temporarily inverting the received bit, only using a ROM of size 25×25 bits. The Matlab simulation result shows that all codes with three or less errors were corrected. Furthermore, the proposed decoding algorithm is simulated based on FPGA EP3C5F265C6 of cyclone III under the flat of Quartus II. The decoding process for the (15,5,7) BCH code used 593 logic unit and the artificial waveform based on VHDL language was shown in Fig. 2. From the waveform we can see when the system clock is 50 MHz, the decoding rate can reach 375 MHz.

Compared with the algebraic decoding algorithm, the improved look-up table algorithm proposed in this study has a much faster decoding speed. For a codeword with the length of $n = 2^m - 1$, the step-by-step decoding algorithm uses a $(m+1)\times 2^m$ bits (80 bit for (15,5,7) BCH code) rom size, while the decoding process is in 2n (30 for (15,5,7) BCH code) clock periods (Zhiyuan *et al.*, 2008). In another hard-decision minimum weight decoder for (15,7,5) BCH code, the decoding time is 1.64 μ sec when using a clock of 25 MHz, that is means the decoding rate is about 0.6 MHz (El-Medany *et al.*, 2002). However, the presented algorithm in present study just uses two clock periods, that is to say when the system clk is 50 MHz, the decoding rate is 375 MHz. It greatly improved the decoding speed.

Based on the look-up table, the number of the syndrome pattern used in the improved algorithm proposed in this study, the traditional look-up table method and Lee's method for (15,5,7) BCH code was shown in Table 1. In the traditional method, the total number of the syndrome pattern is 575, thus the traditional look-up table needs a size of 575×25 = 14375 bits (Lee *et al.*, 2008). The decoding process is to calculate the syndrome pattern and search the same data in the table, so it can achieve the decoding procedure of one codeword in one system clock. In Lee's method, the number of the syndrome pattern is 15, thus the look-up table needs a size of 15×25 = 375 bits. It greatly reduces the memory size, but the decoding process is very complicated. In some situations, the codeword should

Fig. 2: The waveform of the decoding algorithm for (15,5.7) BCH code

Table 1: The number of syndrome patterns of (15,5,7) BCH code used in three different decoding methods

	Syndrome patterns		
Error No.	In the traditional look-up table	In Lee's look-up table	In this proposed look-up table
1	15	5	5
2	105	10	10
3	455	0	10
Total	575	15	25

cyclic shift some bit and repeat the decoding procedure. So, the decoding time is increasing about 10 times (Lee *et al*., 2008). In the proposed algorithm the number of the syndrome pattern is 25, thus the look-up table needs a size of 25×25 = 625 bits. It increases a small amount memory size relative to Lee's table but also saves much memory size relative to the traditional look-up table. However, the decoding process is much more simple than Lee's and it can achieve the decoding procedure of one codeword just in two system clock.

CONCLUSION

An improved decoding algorithm is presented to correct three or less errors in BCH code at a low decoding cost. The look-up table only contains syndrome patterns and corresponding error patterns which have one to three errors occurred in the message block. The error bit can be directly determined used the data in the table and the weight of the syndrome patterns by this algorithm. For (15,5,7) BCH code, only 25×25 = 625 bits memory are needed in the look-up table presented in the study, it can reduce about 95% memory size. In addition the proposed algorithm has no circulation and iteration, this leads to a very fast decoding speed.

ACKNOWLEDGMENT

This study is sponsored by the Aerospace Support Fund and the Harbin Engineering University Fund and the Fundamental Research Fund for the Central Universities.

REFERENCES

Blake, I., C. Heegard, T. Hoholdt and V. Wei, 1998. Algebraic-geometry codes. IEEE Trans. Inform. Theory, 44: 2596-2618.

Chang, H.C., H.P. Lee, T.C. Lin and T.K. Truong 2008. A weight method of decoding the (23, 12,7) golay code using reduced table lookup. Proceedings of the International Conference, May 25-27, Fujian, pp: 1-5.

Chr, C.L., S.L. Su and S.W. Wu, 2004. New step-by-step decoding for binary BCH codes. Proceedings of the 9th International Conference, Nov. 30, IEEE Explore, London, pp: 456-460.

El-Medany, W.M., C.G. Harrison, P.G. Garrell and C.J. Hardy, 2002. VHDL implementation of a BCH minimum weight decoder for double error. Radio Sci., 2: 361-368.

Hung, P., H. Fahmy, O. Mencer and M.J. Flynn, 1999. Fast division algorithm with a small lookup table. Proceedings of the 33rd Asilomar Conference, Oct. 24-27, Pacific Grove, CA., pp: 1465-1468.

Lee, H.P., H.C. Chang, T.C. Lin and T.K. Truong, 2008. A weight method of decoding the binary bch code. Intell. Syst. Des. Appl., 3: 545-548.

Ozkan, A. and E.M. Ozkan, 2002. A different approach to coding theory. J. Applied Sci., 2: 1032-1033.

Penzhorn, W.T., 1993. A fast algorithm for the decoding of binary BCH codes. Proceedings of the IEEE South African Symposium Communication and Signal Processing, Aug. 6, Jan Smuts Airport, South Africa, pp: 63-64.

Peterson, W.W., 1960. Encoding and error-correction procedures for the bose-chaudhuiri code. IEEE Trans. Inform. Theory, 6: 459-470.

Seddiki, A., M. Djebbouri and A. Taleb-Ahmed, 2008. PAPR reduction based on weighted ofdm with product block codes for wireless communication. J. Applied Sci., 8: 4440-4444.

Xiaobei, L., L. Chao, C.T. Hiang and S.N. Koh, 2007. A simplified step-by-step decoding algorithm for parallel decoding of reed-solomon codes. IEEE Trans. Commun., 55: 1103-1109.

Zhiyuan, X., L. Na and L. Lele, 2008. New decoder for triple-error-correcting binary BCH codes. Proceedings of the 3rd IEEE Conference, Aug. 01, Singapore, pp: 1429-1429.