Analysis and Implementation of BCH Codec using PGZ Algorithm

Mahadevan A¹, Priyadharshini A²

¹Assistant Professor, Dept of Electronics and Communication Engineering. SSIET, Chennai. ²PG Scholar, Sree Sastha Institute of Engineering and Technology, Chennai.

Abstract

Data corruption during the transmission and reception of data because of noisy channel medium is the most common problem faced in digital communication system; it is hard to get the reliable communication. Thus, to get the error free communication we need Error correction code. BCH codes are used as a baseline for many recent Error Correcting Codes. BCH code is used to correct multiple random errors. This paper discusses, performance analysis of different types of (15, 7) BCH Decoder. The main aim of this project is to design a Single-bit Error Correcting and Double-Adjacent Error Correcting (SEC-DAEC) parallel decoder that corrects double adjacent and single bit errors in parallel and serially corrects multiple bit errors using Peterson Gorenstein Zierler (PGZ) algorithm. Simulation was carried out by using ModelSim - Altera 6.4a starter edition. Synthesis was successfully done by using Quartus II.

Keywords

BCH codes, SEC-DAEC decoder, Error correcting code (ECC), Verilog, BCH Encoder, and BCH Decoder.

1. Introduction

In the present digital communication systems, it is highly possible that the data or message get corrupted during transmission and reception through a noisy channel medium. The environmental interference and the physical defects in the medium are the main causes of the data or message corruption in the communication medium, which leads to the injection of random bits into the original message and corrupt the original message. To have a reliable communication through noisy medium that has an unacceptable Bit Error Rate (BER) and low Signal to Noise Ratio (SNR), ECC is used. ERROR control codes (also known as error correcting codes, ECCs) have been frequently used to improve the dependability of a memory system [1], [2].

The error correction is based on mathematical formulas, which are used by ECC. Error correction is taken place by adding parity bits to the original message bits during

transmission of the data. Because of the addition of parity bits to message bits makes the size of the original message bits longer. Now this longer message bits is called codeword. This codeword is received by the receiver at destination, and could be decoded to retrieve the original message bits. ECC are used in most of the digital applications, space and satellite communication and cellular telephone networks. There are many types of error correction codes are used in present digital communication system are based on the type of error expected, the communication medium expected error rate, and weather retransmission is possible or not. Some of the error correction codes, which are widely, used these days, BCH, Turbo, Reed Solomon. These codes are different from each other in their complexity and implementation.

The BCH code is one of the best-known and widely used multiple-bit error correcting codes [1], [2]. Multiple-bit error correction of a BCH code needs a low-speed serial decoding process. BCH codes can be decoded faster by parallelizing the serial operations [6], [7], but parallelization incurs in a large hardware overhead, particularly for long information bit length. There are few multiple-bit error correcting codes that can be decoded in parallel, e.g., product codes and some lowdensity parity check (LDPC) codes, such as orthogonal Latin square (OLS) codes [1], Euclidean geometry LDPC (EG-LDPC) codes [9] and difference-set cyclic codes (DSCC) [10]. However, they require longer check bits than BCH codes. To resolve these issues, Wilkerson [11] has proposed a high speed decoding scheme for the BCH code. This scheme utilizes parallel decoding when no error or a single-bit error occurs, and serial decoding when multiple-bit errors occur. As single-bit errors occur more often, this scheme achieves highspeed decoding for most errors.

At the encoder side, binary bits are encoded by appending some parity bits with message bits. The combination of parity bits and message bits are called as codeword. At the decoder side, codeword will be received and error detection and error correction process will be carried out. This process is known as decoding. If error occurs in received codeword, the error will be corrected and original message bit is retrieved. Hence it improves reduction in error rate and it will improve the performance of BCH CODEC.

A. BCH Codes

BCH codes is an acronym for Bose, Roy Chaudhuri, Hocquenghem, invented in 1960s and today they are used as a baseline for many recent ECC. BCH codes are subset of the block codes. BCH codes are belongs to a power full class of multiple error correcting codes. BCH codes are based on welldefined mathematical properties. These mathematical properties are based on the GF or finite fields. The BCH code is a cyclic code, and can be decoded serially. However, the high-speed parallel decoding of a BCH code incurs in a large hardware overhead [6] have proposed a BCH decoding scheme in which only some operations are partially parallelized, and overall, it is slower than fully parallelized decoders. The finite field has the property that any arithmetic operations on field elements always have results in the field only. To provide an excellent error correcting capability, the generator polynomial of the BCH codes has carefully specified roots. With a generator polynomial of g (x), a t error correcting cyclic codes is the binary BCH codes, with a condition that g (x) must be the least degree polynomial over GF. The block length of the BCH code, constructed over GF (2m) is given by the equation n = 2m - 1. BCH codes are cyclic codes, and the degree r of the generator polynomial of a (n, k) is given by the formula $k = 2^m - 1$ - r. In a block codes the codeword is a combination of an information bits and parity bits.

2. BCH Encoder

The encoder design used in this project is most commonly used in the modern digital communication system. This encoder design is almost common to all the BCH code architecture, which uses the linear feedback shift register for polynomial division. The encoder procedure is as follows

n- Block length k- Message bit t- Error correcting code n = 15 k = 7 t = 2

Step 2: Message polynomial

$$u(X) = U_{K-1} X^{K-1} + U_{K-2} X^{K-2} + \dots + U_1 X + U_0$$

(2.1)

Step 3: Generator polynomial

$$g(X) = g_{N-K}X^{N-K} + g_{N-K-1}X^{N-K-1} + ... + g_1X + g_0$$
(2.2)

for t=2

$$g(x) = LCM\{\alpha_1(X), \alpha_3(X)\}$$
(2.3)

for t=3

$$g(x) = LCM\{\alpha_1(X), \alpha_3(X), \alpha_5(X)\}$$
(2.4)

Step 4: Encoding

a) Parity bit generation

 $r(x) = X^{N-K} mod g(x)$ (2.5)

b) Codeword generation

$$c(X) = X^{N-K} u(X) + r(X)$$

(2.6)

where

r(x) is the parity bits polynomial

- c(x) is the codeword polynomial
- u(x) is the message polynomial and
- g(x) is the generator polynomial





BCH codes are implemented as cyclic code. As a result the logic which implements encoder and decoder is controlled into shift register circuits. With the help of cyclic code properties the remainder r(x) can be calculated in the linear

(n-k) stage shift register with the feedback connection to the coefficient of generator polynomial.

In this (15, 7) BCH encoder, binary data of 7 bits are encoded into15 bit codeword using 8 parity bits. Parity bits are generated using generator polynomial by polynomial division.

3. BCH Decoder

The decoding algorithm for BCH codes consist of following steps.

- Compute syndromes
- Determine the error locator polynomial using PGZ algorithm
- Find error location using Chine search algorithm
- Error correction

The 15 bit received data is given as an input to the parallel to serial shift register; the obtained serial output will be used as an input to compute syndromes s(x). If s(x) = 0, the transmission is error free. Otherwise, transmitted message will be in error. This entire process is known as error detection process. The error correction process includes PGZ algorithm and chine's search algorithm. PGZ algorithm accepts syndrome as an input and computes error locator polynomial.



Fig. 2 Block diagram of (15, 7) BCH Decoder

The polynomial can be further used to find the location of the error. Using chine's search algorithm error location is determined. Using the error location information, errors will be corrected by simply flipping (converting 1 to 0 and 0 to 1)

location in received 15 bit code word. The 15 bits corrupted codeword is the input to the decoder module. The task of the decoder is to locate and to correct the errors in the corrupted codeword and retrieve the original message.

4. Results And Discussion

Simulation Results of (15, 7) BCH Encoder and Decoder

The Fig.3 shows simulation waveform of (15, 7) BCH Encoder. As seen in fig.3 when the enable pin is high, the input is loaded to encoder and all other intermediate signals are set to zero. Using this 7 bit binary inputs the parity bits (8 bits) were calculated and these parity bits are appended to the original message bits to obtain a 15 bit codeword or encoded data. The same process repeats for all other input bits as well.



Fig.3 Simulation results for (15, 7) BCH Encoder



Fig.4 Simulation results for (15,7) BCH Decoder

Fig.4 shows simulation waveform of (15, 7) BCH Decoder. As seen in fig.4 when the reset and load pin is high, received codeword is loaded to decoder. At the decoder side the received 15 bit codeword is given as input for syndrome generator. If no error in received codeword syndrome output is zero. If the syndrome output is non zero then the received codeword has some errors. Hence it is detected and corrected using PGZ algorithm and chine's search algorithm.

Synthesis Results of (15, 7) BCH Encoder and Decoder

The fig. 5 and fig.6 shows RTL view of both BCH Encoder and BCH Decoder. Synthesis was successfully done by using Quartus II.



Fig.5 RTL schematic of (15, 7) BCH Encoder



Fig.6 RTL schematic of (15, 7) BCH Decoder

5. Conclusion And Future Work

This project covers the detailed explanation about the necessity of Error correcting code along with the comparison of various error correcting codes and high speed (15, 7) BCH

code encoder and decoder design. The previous chapters discuss the design technique of encoder and decoder, and the behavior of the designs is described using Verilog. If any 2 bit error in any position of 15 bit codeword, it can be detected and corrected. The decoder is implemented using the PGZ and chine search algorithm. Simulation is carried out by using ModelSim and synthesis is carried out using Quartus. The (15, 7) BCH codec can be further implemented in hardware and also performance analysis of different BCH decoder can be carried out.

6. References

- Ibe E., Taniguchui H., Yahagi Y., Shimbo K., and Toba T. (2010) 'Impact of scaling on neutron- induced soft error in SRAMS from a 250 nm to design rule', IEEE Trans. Electron Devices, Vol. 57, No. 7, pp. 1527-1538.
- [2] Liu S.F., Reviriego P., and Maestro J. A. (2012) 'Efficient majority logic fault detection with difference – set codes for memory applications', IEEE Trans. Very large scale integr. VLSI Syst., Vol. 20, No. 1, pp. 148-156.
- [3] Mohammed S. J. and Abdulsada H. F. (2013) 'FPGA implementation of 3 bits BCH error correcting codes', International journal of computer applications, Vol. 71, No. 7, pp. 35–42.
- [4] Neale A. and Sachdev M. (2013) 'A new SEC-DED error correction code subclass for adjacent MBU tolerance in embedded memory', IEEE Trans. Device Mater. Reliab. , Vol. 13, No. 1, pp. 223-230.
- [5] Radaelli D., Punchner H., Wong S., and Daniel S. (2005) 'Investigation of multi bit upset in a 150 nm technology SRAM device', IEEE Trans. Nucl. Sci., Vol. 52, No. 6, pp. 2433-2437.
- [6] Reviriego P., Argyrides C., and Masetro J. A. (2012) 'Efficient error detection in double error correction BCH codes for memory applications', Microelectron. Reliab., Vol. 52, No. 7, pp. 1528-1530.
- [7] Rohith S. and Pavithra S. (2013) 'FPGA implementation of (15, 7) BCH encoder and decoder for text message', IJRET, Vol. 02, No. 09, pp. 209-214.
- [8] Satoh S., Tosaka Y., and Wender S. A. (2000) 'Geometric effect of multiple Bit soft errors induced by cosmic ray neutrons on DRAM's', IEEE Electron Dev. Lett., Vol. 21, No. 6, pp. 301-312
- [9] Violante M., Sterpone L., Manuzzato A., Gerardin S., Rech P., Bagatin M., Paccagnella A., Andreani M., Gorini G., Pietropaolo A., Cardarilli G., Pontarelli S., and Forst C. (2007) 'A new hardware /software platform and a new 1/E neutron source for

soft error studies :FPGAs at the ISIS facility', IEEE Trans.Nucl. Sci., Vol. 54, No. 4, pp. 1184-11889.

- [10] Wang Z. (2013) 'Hierarchical decoding of double error correcting codes for high speed reliable memories', Proc. ACM/EDAV/IEEE Des. Autom. Conf., pp. 1-7.
- [11] Wilkerson C., Alamaldeen A.R., Chisti Z., Wu W., Somasekhar D., and Lu S. (2010) 'Reducing cache power with low cost, multi bit error correcting codes', in Proc. Annu. Int. Symp. Comp. Archit., pp. 83-93.