



INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

ERROR DETECTION USING BINARY BCH (255, 215, 5) CODES

Sahana C*, V Anandi

*M.Tech, Dept of Electronics & Communication, M S Ramaiah Institute of Technology, Bangalore, India
Associate Professor, Dept of Electronics & Communication, M S Ramaiah Institute of Technology,
Bangalore, India

ABSTRACT

Error-correction codes are the codes used to correct the errors occurred during the transmission of the data in the unreliable communication mediums. Error detection is the detection of errors caused by noise or other impairments during transmission from the transmitter to the receiver. The idea behind these codes is to add redundancy bits to the data being transmitted so that even if some errors occur due to noise in the channel, the data can be correctly received at the destination end. The Bose, Ray- Chaudhuri, Hocquenghem (BCH) codes are one of the powerful error-correcting codes. This paper describes the design and simulation of (255, 215, 5) BCH Encoder and Syndrome Calculation circuitry using VHDL for reliable data transfer in AWGN channel with error correcting capability of $t=5$. The digital logic implementation of binary encoding of BCH (255, 215, 5) of length $n=255$ over $GF(2^8)$ with primitive polynomial $1+x+x^2+x^7+x^8$ is organized into Linear Feedback Shift Registers (LFSR). The proposed syndrome block is used to optimize the hardware consumption required for the design and implementation.

KEYWORDS: BCH Encoder, LFSR, Syndrome Calculator

INTRODUCTION

Claude Shannon proposed the theorem of Channel capacity stating that, "Channel capacity is the maximum rate at which bits can be sent over the channel with arbitrarily good reliability"[1]. According to Channel Coding theorem, "The error rate of data transmitted over a band-limited noisy channel can be reduced to an arbitrarily small amount if the information rate is lower than the channel capacity" [2]. Error correcting codes are used in satellite communication, cellular telephone networks, body area networks and in most of the digital applications. There are different types of error correcting codes based on the type of error expected, expected error rate of the communication medium, and whether re-transmission is possible or not. Few of them are BCH, Turbo, Reed Solomon, Hamming and LDPC. These codes differ from each other in their implementation and complexity.

Error Correction Codes are required to increase the reliability of binary transmission (or storage) system. To have a reliable communication through noisy medium that has an unacceptable bit error rate (BER) and low signal to noise ratio (SNR), we need to have Error Correcting Codes which is based on proven mathematical formulas. Error correction is taken place by adding parity bits to the original message bits during transmission of the data. Error correcting codes have a wide range of applications in different fields like digital data communications, memory system design, and fault tolerant computer design among others.

Error detection is the detection of errors caused by noise or other impairments during transmission from the transmitter to the receiver. It uses the concept of redundancy, which means adding of extra bits for detecting errors at the destination. In error correction the receiver can use any of the error-correcting code, which can automatically corrects certain errors and enables reconstruction of the original data.

MATERIALS AND METHODS

I. BCH CODES

BCH abbreviation stands for the discoverers, Bose and Chaudhuri (1960) and independently Hocquenghem (1959). BCH codes are cyclic codes which is a subclass of linear block codes. A linear block code is said to be a cyclic code when it obeys the cyclic property. Cyclic codes[4] form a subclass of linear block codes. This class of codes is a remarkable generalization of the Hamming codes for multiple error correction. The most common binary BCH codes

are characterized for any positive integers m (equal to or greater than 3) and the number of errors detected and corrected t by the following parameters:

$$\begin{aligned} \text{Block length: } n &= 2^m - 1 \\ \text{Number of message bits: } k &\geq n - mt \\ \text{Minimum distance: } d_{\min} &\geq 2t + 1 \end{aligned}$$

Each BCH code is a t-error correcting code in that it can detect and correct up to t random errors per code word. The Hamming single error correcting codes can be described as BCH codes. The BCH codes offer flexibility in the choice of code parameters, namely, block length and code rate. Furthermore, for block lengths of a few hundred bits or less, the BCH codes are among the best known codes of the same block length and code rate.

BCH ENCODER DESIGN

The BCH code operates in Galois Field. It can be defined by two parameters that are the length of code words (n) and the number of errors to be corrected t. A t-error correcting BCH code is capable of correcting any combination of t or fewer errors in a block of $n = 2^m - 1$ digits. The code words are obtained by taking the remainder after dividing a polynomial representing the information bits by a generator polynomial.

The generator polynomial is selected to give the code its characteristics. All code words are multiples of the generator polynomial. The generator polynomial is the polynomial of lowest degree over GF(2) with $\alpha, \alpha^2, \alpha^3, \dots, \alpha^{2t}$ as roots [$g(\alpha^i) = 0$ for $1 \leq i \leq 2t$]. The generator polynomial is the least common multiple of the minimal polynomials of each α^i term, where α is a primitive element in GF(2^m). Let $\phi_i(x)$ be the minimal polynomials of α^i , then the generator polynomial g(x) must be,

$$G(x) = \text{LCM} \{ \phi_1(x), \phi_2(x), \phi_3(x), \dots, \phi_{2t}(x) \}$$

A simplification is possible because every even power of a primitive element has the same minimal polynomial as the odd power of the element i.e. $\alpha^{2i} = (\alpha^i)^2$, where $I = i * 2^l \geq 1$

So the generator polynomial can be reduced as

$$G(x) = \text{LCM} \{ \phi_1(x), \phi_3(x), \phi_5(x), \dots, \phi_{2t-1}(x) \}$$

An irreducible polynomial g(x) of degree m is said to be primitive if and only if it divides polynomial form of degree n, $X^n + 1$ for $n = 2^m - 1$.

For (255, 215) BCH code, let α be a primitive element of GF (2^8). We get the minimal polynomials of $\alpha, \alpha^3, \alpha^5, \alpha^7, \alpha^9$ as,

$$\begin{aligned} \phi_1(x) &= 1 + x + x^2 + x^7 + x^8 \\ \phi_3(x) &= 1 + x^2 + x^3 + x^4 + x^6 + x^7 + x^8 \\ \phi_5(x) &= 1 + x + x^4 + x^5 + x^6 + x^7 + x^8 \\ \phi_7(x) &= 1 + x^2 + x^3 + x^7 + x^8 \\ \phi_9(x) &= 1 + x + x^3 + x^4 + x^5 + x^6 + x^8 \end{aligned}$$

For t=5 error correcting, BCH code of length $n = 2^8 - 1 = 255$ is generated by

$$G(x) = \text{LCM} [\phi_1(x), \phi_3(x), \phi_5(x), \phi_7(x), \phi_9(x)]$$

$$\text{i.e. } G(x) = 1 + x + x^4 + x^7 + x^9 + x^{11} + x^{12} + x^{15} + x^{19} + x^{22} + x^{24} + x^{31} + x^{32} + x^{33} + x^{34} + x^{38} + x^{40}$$

The highest degree of the polynomial is 40 i.e. $(n-k = 255-215 = 40)$, thus the code is a (255, 215) cyclic code.

BCH encoder is implemented with serial linear feedback shift register architecture.

BCH code words are encoded as,

$$c(x) = m(x).x^{n-k} + b(x)$$

where b(x) denotes the remainder polynomial of dividing f(x) by g(x).

$$c(x) = c_0 + c_1 x + \dots + c_{n-1} x^{n-1}$$

$$i(x) = i_0 + i_1 x + \dots + i_{k-1} x^{k-1}$$

$$b(x) = b_0 + b_1 x + \dots + b_{n-k-1} x^{n-k-1}$$

where c(x) is the codeword polynomial, i(x) is the message polynomial, b(x) is the parity polynomial.

The remainder polynomial b(x) can be obtained in a linear (n-k) stage feedback connections corresponding to the coefficients of the generator polynomial.

$$g(x) = 1 + g_1 x + \dots + g_{n-k-1} x^{n-k-1} + x^{n-k}$$

Such a circuit is shown in the figure 1.

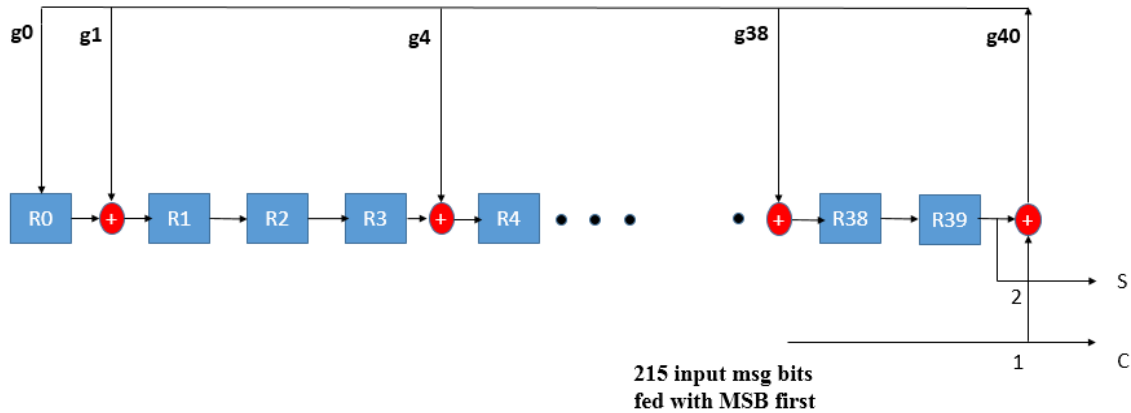


Figure.1 BCH Encoder using LFSRs

On the encoder side, systematic encoding has been used. In systematic encoding, the message bits will be transmitted in unaltered form and the parity bits are transmitted following the information bits.

The encoder which is shown in Figure 1 operates as follows

For clock cycles 1 to k, the information bits are transmitted in unchanged form with switch S2 in position 2. Meanwhile the parity bits are calculated in the LFSR with switch S1 on.

For clock cycles k+1 to n, the parity bits are transmitted with switch S2 in position 2. This time the feedback switch S1 will be in the on position.

To improve the speed of encoding the presence of the switch S2 is eliminated in the VHDL code.

That is the code word output will be equal to the incoming message bits when S1 is on and the code word output will be equal to the parity bits when switch S1 is open.

SYNDROME CALCULATION

The syndrome calculator is the first module at the decoder, the design of this module is almost same for all the BCH decoder architectures. The input to the syndrome module is the received codeword. The received polynomial may be corrupted with error pattern $e(x)$ as:

$$r(x) = c(x) + e(x)$$

where the received codeword is

$$r(x) = r_0 + r_1 x + r_2 x^2 + \dots + r_{n-1} x^{n-1}$$

Transmitted codeword is given by:

$$c(x) = c_0 + c_1 x + c_2 x^2 + \dots + c_{n-1} x^{n-1}$$

The error pattern is:

$$e(x) = e_0 + e_1 x + e_2 x^2 + \dots + e_{n-1} x^{n-1}$$

Syndrome S_i can be computed as:

$$S_i = r(\alpha^i) = r_0 + r_1 \alpha^i + r_2 \alpha^{2i} + \dots + r_{n-1} \alpha^{(n-1)i} \text{ where } 1 \leq i \leq 2t - 1.$$

For hardware implementation, syndrome components can be computed using linear feedback shift registers as $S_i = r(x)/\phi(x)$

For BCH (255, 215, 5) the $2t$ syndromes i.e. 10 syndromes are calculated as:

$S_1 = r(\alpha)$	$S_3 = r(\alpha^3)$
$S_2 = r(\alpha^2) = S_1^2$	$S_6 = r(\alpha^6) = S_3^2$
$S_4 = r(\alpha^4) = S_2^2$	$S_5 = r(\alpha^5)$
$S_8 = r(\alpha^8) = S_4^2$	$S_{10} = r(\alpha^{10}) = S_5^2$
$S_7 = r(\alpha^7)$	$S_9 = r(\alpha^9)$

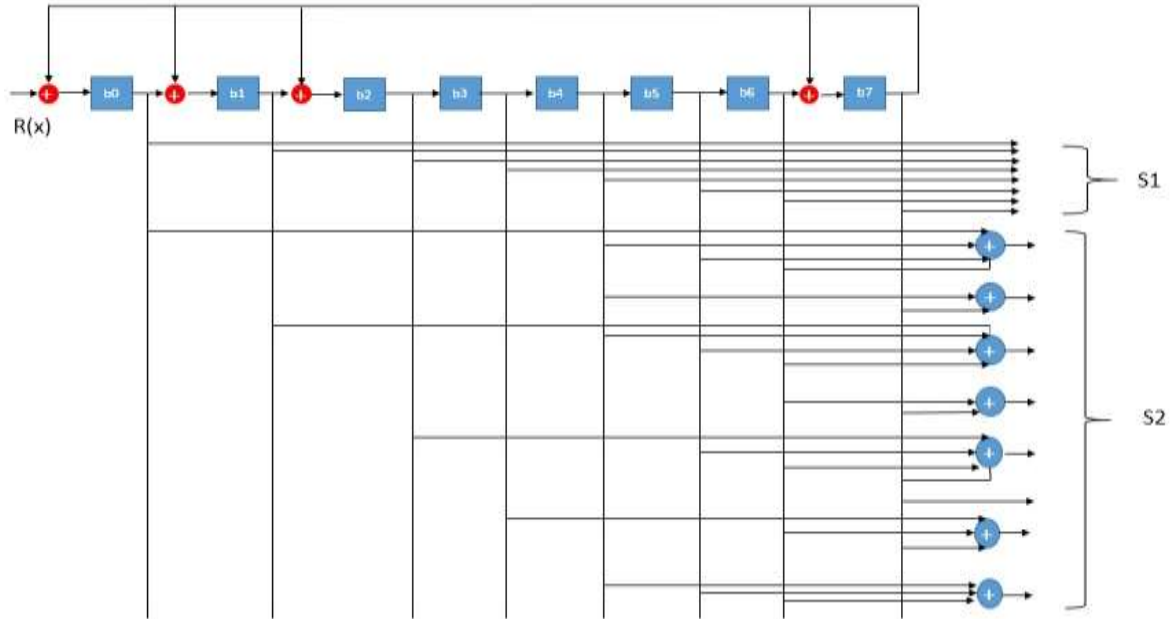


Figure.2 Implementation of Syndromes S1 and S2

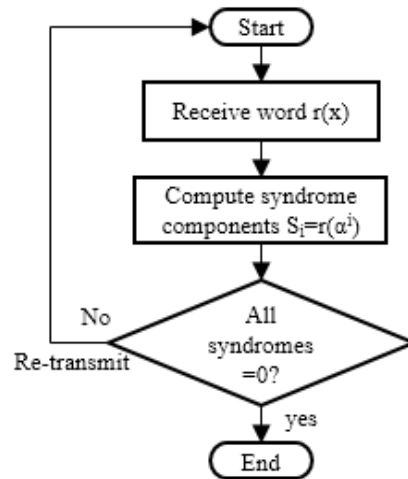



Figure.3 Flowchart for Error detection

RESULTS AND DISCUSSION

The proposed BCH(255,215,5) Encoder and Syndrome calculation based on Minimal polynomial method have been designed using VHSIC Hardware Description Language (VHDL) and simulated using ModelSim 10.1c. The results were also verified in MATLAB 7.8.0.

Figure 4 and Figure 5 shows the simulation results of BCH encoder and Syndrome Calculation respectively. If the transmitted and the received codewords are the same then the syndromes will be zero. Here in this case the received codeword as erroneous is discussed. The received 255 bit encoded data given as input to the syndrome calculation circuit. Due to the presence of error the syndrome value will be a non-zero. Once the error is detected, re-transmission of data is requested. For error correction, Berlekemp Massey Algorithm and Chien search algorithm can be employed

AUTHOR BIBLIOGRAPHY

	<p>Sahana C Obtained her B.E in 2012 in Electronics & Communication from UBDTCE, Davangere, Karnataka. Currently pursuing her M.Tech in Digital Electronics & Communication from M S Ramaiah Institute of Technology, Bangalore, Karnataka. Email: sahana0007@gmail.com</p>
	<p>V Anandi Currently working as Associate Professor in Dept of Electronics & Communication at M S Ramaiah Institute of Technology, Bangalore, Karnataka. Her research areas include VLSI Design. Email: anandi.v@msrit.edu</p>