Error Control Circuit And Its Implementation On Radio Link Enhancement.

Sirath. A, Sharqa Ekram, Sinky Kumari
Department of Telecommunications
Dayanada Sagar College of Engineering

Abstract: This paper proposes a method on designing a circuit for Error Correction and implementing it for Enhancement of a Radio Link. BCH codes are preferred for the method of Forward Error Correction (FEC) as using FEC removes the disadvantage of Error Detection Method in terms of Bandwidth utilization. Theory on "designing a circuit of one bit error correcting capability with code size of 7 bits" already exists. This paper proposes an extension of the error correcting capability of the circuit up to 7 bits with code size of 255 bits, using BCH codes. Code size and Number of errors are parameters which define any BCH Code.

I. Introduction

In information theory and coding which has various applications in the field of telecommunication, **error detection and correction** or **error control** are techniques that enable reliable delivery o of digital data over unreliable communication channels. Many communication channels are subject to channel noise, and thus errors may be introduced during transmission from the source to a receiver. Error detection techniques allow detecting such errors, while error correction enables reconstruction of the original data. **BCH** In coding theory the **BCH codes (***Bose, Chaudhuri, and Hocquenghem***)** form a class of cyclic error-correcting codes that are constructed using finite fields. BCH codes are used in applications like satellite communications, various types of compact disk players, DVDs , disk drives, solid-state drives and also two-dimensional bar codes Implementation of only **error detection methods** on the receiver side, leads to disadvantages of **bandwidth utilization**. In order to overcome this effect the use of error controlling methods on the receiver side results in proper **link enhancement.** BCH codes operate over finite fields or Galois fields. BCH codes can be defined by parameters that are code size denoted as **'*n*'** and the error correcting capability denoted as't'. The importance of BCH codes stems from the fact that they are capable of correcting all random patterns errors by decoding algorithm that is simple and easily implemented. This project considers the use of a FPGA (*Field Programmable Gate Array.* A **field-programmable gate array** (**FPGA**) is an integrated circuit designed to be configured by a customer or a designer after the manufacturing is done—hence "field-programmable". FPGAs contain programmable logic components which are basically called "logic blocks", and a hierarchy of reconfigurable interconnects that allow the blocks to be "wired together"—somewhat like many (changeable) logic gates that can be inter-wired in (many) different configurations. Logic blocks can be configured to perform complex combinational functions, or merely are just few simple logic gates like AND and OR and XOR . In most FPGAs, the logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory.

This paper is our study on implementation of BCH coding in FPGA. For **two bit correction**, BCH codes need to generate **16 bits parity** for **239 bits** data, and has the length word or **code size** as **255 bits.** This mode is usually called **(255,239)** BCH code.

II. Basic Theory

A. BCH ENCODER:

[1]The BCH codes are a class of cyclic codes whose generator polynomial is the product of distinct minimal polynomials corresponding to

$$\alpha, \alpha 2, \ldots, \alpha 2t. \quad (1)$$

where $\alpha \in GF(2m)$ is a root of the primitive polynomial $p(x)$. Let $mi(x)$ be the minimal polynomial of αi. Let

$$c(x) = c0 + c1\,x + c2\,x2 + \ldots + c\,n\text{-}1\,xn\text{-}1 \quad (2)$$

be a code polynomial with coefficients from $GF(2)$. If $c(x)$ has $\alpha, \alpha 2, \ldots, \alpha 2t$ as its roots, $c(x)$ is then divisible by the minimal polynomials m1(x), m2(x), …, m2t(x) of $\alpha, \alpha 2, \ldots, \alpha 2t$. The generator polynomial g(x) of the t-error-correcting BCH code of block length n is,

n = 2m – 1 & rate k/n (3)

is the lowest degree polynomial over GF(2). Thus, the generator polynomial of the code must be the least common multiple of the minimal polynomials.

That is,

g(x)=LCM{m1(x),m2(X),……,m2t(x)(4)

Encoding is done using shifters and adders. The coefficients of the generator polynomial are given as inputs to the adder along with the output from the previous shift register.

B. BCH DECODER:

It recovers a binary message vector from a BCH code vector. Input is the Binary code vector and the first output is the corresponding Binary message Vector. The second output is the Number of error detected during decoding of the Code which has following processes. Decoding involves three processes. Syndrome Calculation: To find error locator polynomial and its roots which is done by method using BERLEKAMP MASSEY algorithm and correction of errors in received vectors.

C. BERLEKAMP MASSEY algorithm-

Berlekamp's algorithm is a well-known method for factoring polynomial over finite fields. It is used for decoding of BCH code using a LFSR. LFSR is a feedback register which generates periodic sequence using characteristic polynomial. LFSR is a register whose input bit is a linear.It will find shortest LFSR for a given Binary Output Sequence.It will also find the minimal polynomial of a Linearly Recurrent Sequence. In Massey Algorithm we built LFSR that produce entire sequence of {S1, S2, S2t}, by successfully modifying an existing LFSR.

The error locator polynomial is given by;

e(x)=∑Xi^j; (5)
$0 \leq j \leq 2t$; (6)
$0 \leq i \leq n-1$. (7)
To determine the location of the errors the above mentioned equation has to be solved in order to find the roots. These roots will determine the location of errors in the sequence of received vector.

D. Syndrome Calculation

Syndrome digits are linear combination of the error digits. A syndrome provides information about error digits. Therefore they are used in error detection and also error correction at certain levels. The first step in decoding a ***t* -error - correction** BCH codes is to compute the syndrome components may be ***2t*** syndrome components *s1 , s2 ,..., s2t*.These obtained by substituting the field elements α, α*2 , ... ,* α*2t* into the received polynomial *r(x)*.Thus, the *i*th component of the syndrome is

Si=r (α^i); (8)
S=∑(rk) α^jk; $0 \leq k \leq n-1$; j=1,2,....,2t; (9)

The syndrome components are a function of the field elements of GF(2m).

The next step in decoding is to find the error locator polynomial and its roots. This is done using BERLEKAMP MASSEY algorithm.

E. Galois Fields:

It is a finite field which contains finite number of elements. The elements of the field are the coefficients of the polynomial. Error control codes rely to a large extent on powerful and elegant algebraic structures called finite fields. A field is essentially a set of elementin which it is possible to add, subtract, multiply and divide field elements and always obtain another element within the set. A field *F* is a non-empty set of elements with two operators usually called addition and multiplication, denoted "+" and "*" respectively.

F. The Error-Locator Polynomial:

Suppose $v \leq t$ errors actually occur, let the error locator polynomial ρ(x) be

$\rho(x)=\rho_0+\rho_1 x+\ldots\ldots+\rho_v x^v$ (10)

ρ_i, $0 \leq i \leq v$, are closely related to the syndrome components *sj* , $1 \leq j \leq v+1$.

III. EXPERIMENTAL RESULTS

Consider the (255,239) BCH code. Let α be a primitive element of the Galois field $GF(2^8)$. If $mi(x)$, $i = 1,2, \ldots, 254$, denote the minimal polynomials of α_i, which are the elements of $GF(2^8)$,

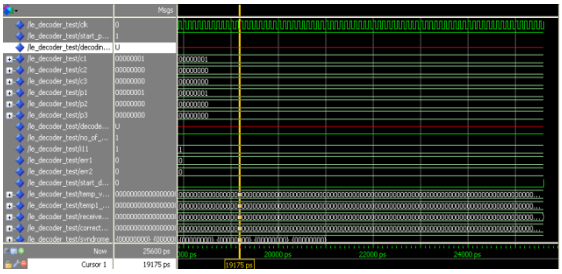
The generator polynomial of the (255,239) BCH code can be given by

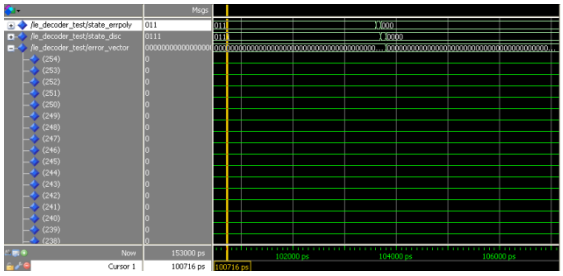
$$G(x)=LCM\{m_1(x),m_2(x),\ldots..,m_{254}(x)\} \quad (11)$$

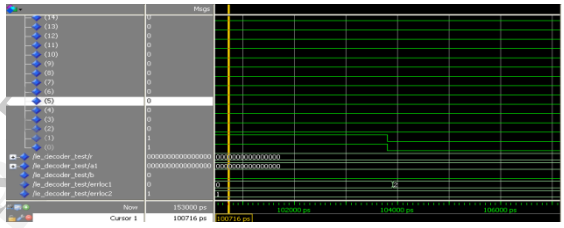
The encoding circuit for the (255,239) BCH code is easily implemented as shown in Figure. The output of figure is a serial bit of 255 bit data generated by BCH encoder.

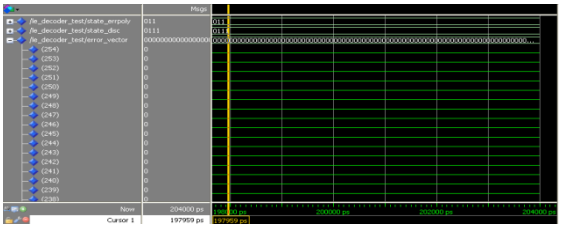
Decoding of the received vector involves syndrome calculation, Error Position polynomial calculation ,Error Position retrieved and error value calculation, Error Correction. If no error in $r(x)$ exist, the decoder generates no syndromes. Therefore, the error-locator polynomial simply becomes 1. The error location is calculated using Beerlekamp Massey Algorithm.Simulation has been done in modelsim Altera 6.6c(Quartus II 10.1) starter edition, before it is implemented in FPGA.The result show that the circuits work well, any 2 bit error in any position of 255 bits has been corrected. Our next project is to build 7 bits error correction and BCH code size will be 255 bits.

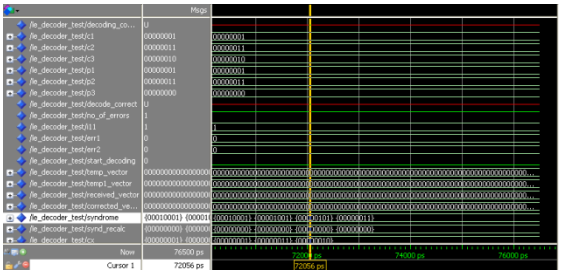
The snapshots shown represent certain simulation results which shows the error controlling for given outputs.

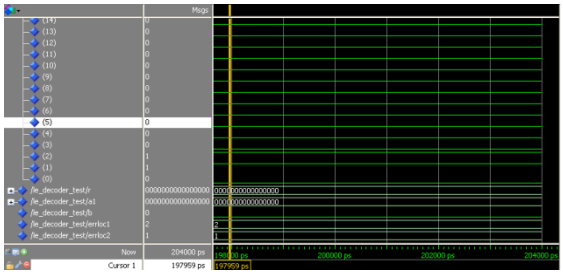












IV. Conclusion

The result show that the circuits work well, any 2 bit error in any position of 255 bits has been corrected. Our next project is to build 7 bits error correction and BCH code size will be 255 bits. It can correct 2 error at the receiver side when the original data gets corrupted by the noise. BCH codes have been shown to be excellent error correcting codes among codes of short lengths. They are simple to encode and relatively simple to decode. Due to these qualities, there is much interest in the exact capabilities of these codes. The speed and device utilization can be improved by adopting parallel approach methods.

7. Acknowledgement

We sincerely thank Mr Vinod. B. Durdi., (Associate Professor, Dept of Telecommunication) for her skillful guidance, constant supervision and timely suggestions. We also thank Dr. Ashwatha, (Head of the Department, Dept. of Telecommunication) for his support.

IV. References

[1] IEEE paper on BCH CODES published Joachim Rosenthal n Eric V. York.

[2] FPGA Implementation of Encoder for (15, k) Binary BCH Code Using VHDL and Performance Comparison for Multiple Error Correction Control based IEEE paper by Panda, A.K.

[3] Closed solution of the *Berlekamp–Massey algorithm based* IEEE paper written by Kraft, C in Dec 1991.

[4] BCH code based multiple bit error correction in finite field multiplier circuits theory based IEEE proposed paper by Mahesh P., Jimson Mathew, A.M. Jabir, Dhiraj Pradhan and S. P. Mohanty.

[5] IEEE paper on Bch Decoder Architectures by Yi-Min Lin, Hsie-Chia Chang, and Chen-Yi Lee.

[6] Book on Circuit Design with VHDL by Volnei A. Pedroni.

[7] Book on Error Correcting Coding by Todd K Moon.

[8] Introduction to Error Correction by Shu Lin.