An Area Efficient (31, 16) BCH Decoder for Three Errors

M. Prashanthi[1], P. Samundiswary[2]

[1]M. Tech, Department of Electronics Engineering, Pondicherry University, Pondicherry, India
[2]Assistant Professor, Department of Electronics Engineering, Pondicherry University, Pondicherry, India

***Abstract*— Bose, Ray- Chaudhuri, Hocquenghem (BCH) codes are one of the efficient error-correcting codes used to correct errors occurred during the transmission of the data in the unreliable communication mediums. This paper presents a low-complexity and area efficient error-correcting BCH decoder architecture for detecting and correcting three errors. The advanced Peterson error locator computation algorithm, which significantly reduces computational complexity, is proposed for the IBM block in the (31,6) BCH decoder. In addition, a modified syndrome calculator and chien search are proposed to reduce hardware complexity. The design methodologies of the proposed BCH decoder models have considerably less hardware complexity and latency than those using conventional algorithms. The proposed (31, 16) BCH decoder over GF (5), leads to a reduction in complexity compared to that of conventional BCH decoder. The enhanced BCH decoder is designed using hardware description language called Verilog and synthesized in Xilinx ISE 13.2.**

Keywords*— BCH decoder, Peterson's algorithm, chien Search, VLSI design.

I. INTRODUCTION

BCH codes are large class of multiple error-correcting codes. The binary BCH codes were discovered in 1959 by Hocquenghem and independently in 1960 by Bose and Ray-Chaudhuri. Later, Gorenstein and Zierler generalized them to all finite fields.

In practice, the binary BCH and Reed-Solomon codes are the most commonly used variants of BCH codes. They have applications in a variety of communication systems: digital subscriber loops, wireless systems, networking communications, and magnetic and data storage systems. Continuous demand for ever higher data rates and storage capacity provide an incentive to devise very high-speed and space-efficient VLSI implementations of BCH decoders. The first decoding algorithm for binary BCH codes was devised by Peterson in 1960.

BCH codes are generally developed in Galois Field (GF).The Galois field was invented by Everest Galois. GF has finite number of elements in it. The theory of Finite field introduced around 18th century, while its importance and the widespread applications widely recognized in recent years. The GF is widely used in number theory, coding theory and cryptography. The mathematical properties within which BCH codes are defined, also represents Galois field[6]. The mathematical operations like additions, subtractions, multiplications and divisions are performed using Finite field theory. The most basic axioms of the finite field are:

a) All the elements in the field forms an Abelian group with additional operator "+".
b) The non-zero elements in the field forms group with multiplication operator " **.**".
c) Multiplications by any non –zero elements is an automorphism of the Additive group.

There exists a Galois field, also known as the finite fields, over the set GF(p) for every prime number p. The finite fields are having p elements with special elements 0 and 1 as the additive and multiplicative identities respectively. The extended field, denoted as GF(p^m), are comprised of the polynomials of degree m−1 over the field Zp. These polynomials are expressed as $a_{m-1}\ x^{m-1}+...+a_1x^1+a_0x^0$ where the coefficients a_i take on values in the set {0, 1, ..., p − 1}.When this extended field is used in coding applications p is commonly 2 and thus the coefficients {a_0, ..., a_{m-1}} are taken from the binary digits {0, 1}. The variable t is defined as an error detecting BCH code of length n over GF (p^m) as the set of all sequences. The code length n is equal to $p^m - 1$.The BCH code with m=1 is called the Reed Solemon code. BCH codes are widely used in error-coding techniques, concatenated either with convolution codes or with other block codes such as Low-density parity check (LDPC) codes. Second generation Digital Video Broadcast (DVB) standards are adopting BCH codes as part of their concatenated coded strategy [7].

BCH codes architecture use field theory as well as polynomial over finite filed. To detect any error has occurred during transmission, a check polynomial is constructed. The BCH code with distance δ over the finite field GF (p^m) is constructed by finding polynomial over GF (p), whose roots include δ consecutive power of y. To correct t number of errors, a BCH code is implemented in ordered form. In this form, a BCH code is obtained by dividing the message polynomial with the generator polynomial and then appending the remainder with the message to form the codeword polynomial. The generator polynomial is formed by taking the

Least-Common Multiple (LCM) of all the minimal polynomials corresponding to the roots. The division is implemented using LFSR architecture.

The objective of this work is to reduce the number of components used for designing the triple error correction for a BCH decoder by incorporating modified structures of various blocks which is described in this paper . The Peterson's algorithm reduces the complexity along with size of the Look Up Tables (LUT) for the IBM block and new factorization method of the chien search block reduces the number of logic gates used to find the error location.

This paper is organized as follows; Section II explains the existing model of the BCH decoder and section III explains the modified BCH decoder and detailed structure of syndrome, IBM and Chien search respectively. Results are analysed in the section IV. Section V is drawn with the conclusion.

II. Existing BCH Decoder Architecture

The BCH decoder normally consists of four modules namely:

- Syndrome Calculator
- Inversion less Berlekamp Algorithm
- Chien Search
- Error Correction

A. *Syndrome Calculator*

The syndrome calculator is the first module of the BCH decoder. The input to this module is the corrupted codeword. The equations for the codeword, received bits and the error bits are given as below.

Codeword equation

$$c(x) = c_0 + c_1x + c_2x^2 + \ldots + c_{n-1}x^{n-1} \qquad \ldots(1)$$

Received bits equation

$$r(x) = r_0 + r_1x + r_2x^2 + \ldots + r_{n-1}x^{n-1} \qquad \ldots(2)$$

Error bits equation

$$e(x) = e_0 + e_1x + e_2x^2 + \ldots + e_{n-1}x^{n-1} \qquad \ldots(3)$$

Thus, the final transmitted data polynomial equation is given as below:

$$r(x) = c(x) + e(x) \qquad \ldots(4)$$

The 1st step at the decodin process is to store the transmitted data polynomial in the buffer register and then to calculate the syndromes S_j. The important characteristic of the syndromes is that it depends only on error location not on transmitted information [5]. The equation of the syndromes is given as follows which define the syndromes $4S_j$ as

$$S_j = \sum_{i=0}^{n-1} r_i \alpha^{i \cdot j} \text{ for } (1 \le j \le 2t). \qquad \ldots(5)$$

B. *Inversion less Berlekamp Massey (IBM)*

The second stage in the decoding process is to find the co-efficient of the error location polynomial using the generated syndromes in the previous stage. The error location polynomial is given as: $\sigma(x) = \sigma_0 + \sigma_1 x + \ldots + \sigma_t x^t$. The relation between the syndromes and the error location polynomial is given as below [1]:

$$\sum_{j=0}^{t} S_{t+i-j}\sigma_j = 0 \text{ (i= 1, ..., t)} \qquad \ldots(6)$$

There are various algorithms used to solve the key equation solver. In this project Inversion less Berlekamp Massey algorithm is used to solve the key equation [2].

Berlekamp Massey Algorithm:

The steps of Berlekamp Massey algorithm is given as below:

1. First step is to calculate error syndromes Sj.
2. Initialize the k = 0, $\Lambda^{(0)}(x) = 1$, L = 0 and T(x) = x
3. Assign k = k + 1 and then the discrepancy $\Delta^{(k)}$ is then calculated as follows:

$$\Delta^{(k)} = S_k - \sum_{i=1}^{L} \Lambda_i^{(k-1)} S_{k-i} \qquad \ldots(7)$$

4. If the value of $\Delta^{(k)}$, 2L $\ge$ equals 0, then go-to step 7.
5. Calculate the $\Lambda^{(k)}(x) = \Lambda^{(k-1)}(x) - \Delta^{(k)} T(x)$
6. Set the value of L = k – L and T(x) is calculated as
   - $T(x) = \Lambda^{(k-1)}(x) / \Delta^{(k)}$
   - Set T(x) = x.T(x).
7. If the value of k< 2t , then go-to step 3 Continue for i = 2t – 1 and then End.

C. *Chien Search*

Chien Search block is the third step of decoding process, which is used to calculate the error location [4].

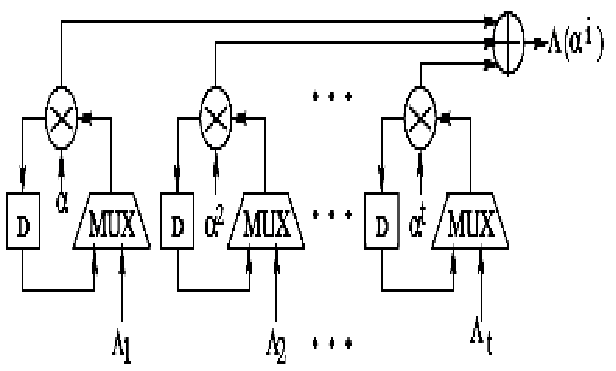

Figure 1: Internal Structure of Existing Chien Search

To locate the roots of the polynomial, chien search block is implemented in the hardware for the BCH decoder. The chien search block has a property that the roots are going to be a power of α, which reduces the evolution of the polynomial for every root[3]. So, the chien search block is used to provide the computational benefits of this step,

$$\Lambda_i^{(j)} = \Lambda_i^{(j-1)} \alpha^i \qquad \ldots(1.7)$$

This property makes the chien search method more superior than other methods

III. Modified BCH Decoder Architecture

A. Syndrome Calculator

The modified syndrome used in this paper is easy to compute and provide a simple and scalable approach. There will be 2t syndromes which can be corrected by the decoder.

$$S_{2t} = (S_t)^2 \qquad \ldots(1.8)$$

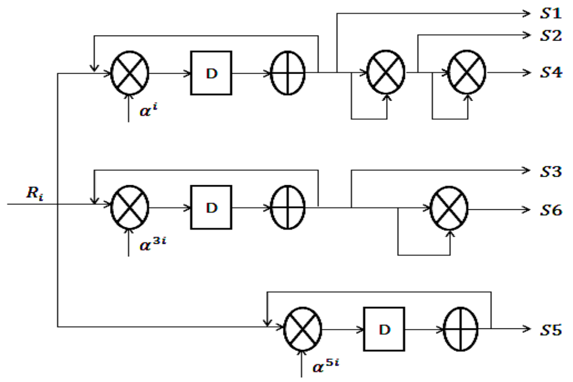

Figure 2: Internal Structure of modified Syndrome Block

Thus, 2t syndromes are computed instead of t syndromes by using the above equation. As the result, area is reduced since the number of components required to calculate the syndromes are reduced. The internal structure of modified syndrome block is shown in Figure 2.

B. IBM

The roots of the three error polynomial are computed directly by transforming the error polynomial into a Peterson error locator computation algorithm. Thus, the more efficient and low complexity Peterson error locator computation algorithm is used for t=3, where $\Lambda(x)$is computed directly. The degree of ($\Lambda(x)$) is equal to the number of errors happened.

For t=3, the matrix (A) given as

$$\begin{bmatrix} 1 & 0 & 0 \\ S_2 & S_1 & 1 \\ S_4 & S_3 & S_2 \end{bmatrix} \begin{bmatrix} \Lambda_1 \\ \Lambda_2 \\ \Lambda_3 \end{bmatrix} = \begin{bmatrix} S_1 \\ S_3 \\ S_5 \end{bmatrix}$$

- If the above matrix is non-singular then det(A)= $S_3 + S_1 S_2$
- If $S_3 = S_1 S_2$ only single error has occurred.
- When $S_3 \neq S_1 S_2$ more than two errors has occurred.
- By solving the above matrix we get:
  - Λ_0=0001
  - $\Lambda_1 = S_1$
  - $\Lambda_2 = S_2 S_3 + S_1 S_4$
  - $\Lambda_3 = (S_1^3 + S_3) + S_1 \Lambda_2$

C. Chien Search

The error position is calculated by using the below equation:

$$\Lambda(\alpha^i) \qquad for(0 \le i \le n-1) \quad \ldots (8)$$

where, $\Lambda(\alpha^i)$ is the error locator polynomial.

For the codeword (n, k) , the error position is calculated as

$$\Lambda(\alpha^{(n-1)}\ \Lambda(\alpha^{(n-2)}, \ldots\ldots\ \Lambda(\alpha^{(0)}) \qquad \ldots (9)$$

If the expression $\Lambda(\alpha^i)$ reduces to zero, then that value is the root and it identifies the error position else the position does not contain any error. The internal structure of modified chien search block is shown in Figure 2.

Factorization Method

In this method, another form of chien search circuit is designed that minimizes large number of the logic gates used in the circuit, by factorizing the error locator polynomial (Λ) such that it is not depicted in its expression.

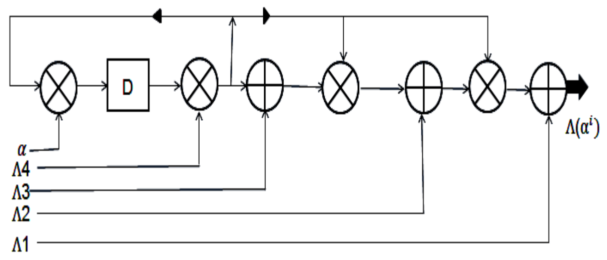

Figure3: Internal Structure of Modified Chien Search

IV. Simulation Results & Analysis

The enhanced (31,6) BCH decoder is designed using Verilog. Then the advanced decoder is simulated and synthesized in Xilinx ISE 13.2.

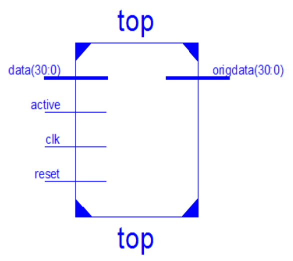

Figure 4: Top View of (31,16) BCH Decoder

Figure 4 illustrates the top view of (31, 16) BCH decoder where ‘data’ of 31bits is the corrupted codeword given as the input to the decoder and the output ‘origdata’ is the recovered 31 bit uncorrupted data.

Internal structure of BCH decoder shown in figure 5 is the integration of syndrome, IBM, chien search and error correction blocks.

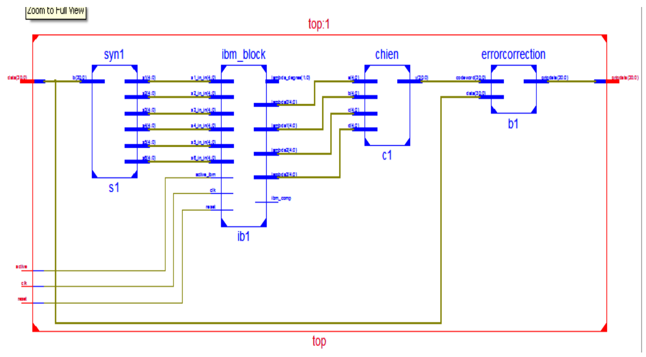

Figure 5: Internal Structure of BCH Decoder

The below figure 6 shows the Register Transfer Logic (RTL) View of the BCH decoder which consists of logical gates required for all the internals bolcks. The timing waveform of BCH decoder is illustrated in figure 7.

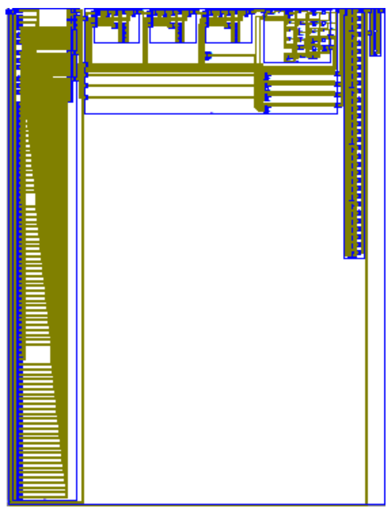

Figure 6: RTL View of BCH Decoder

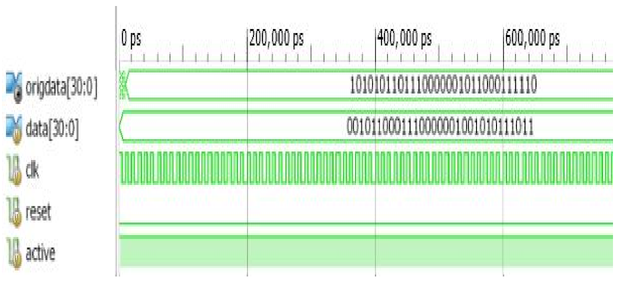

Figure 7: Timing Waveform of BCH Decoder

Table 1: Comparsion of Existing and Propsed (31.16) BCH Decoder

Block	*No of Slices*		*4 Input LUT*		*Delay(ns)*	
	Existing	**Proposed**	**Existing**	**Proposed**	**Existing**	**Proposed**
BCH Decoder	606	399	1040	714	15.94	15.32
Syndrome	49	35	85	61	9.21	9.37
IBM	297	50	559	89	7.1	5.91
Chien Search	162	150	282	267	11.11	12.76

From the above Table 1 it is infered that the existing model occupies more area than the proposed model since it has less no of slices and 4 input Look Up Tables(LUTs) compared to that of existing model

V. Conclusion

An Area Efficient (31, 16) BCH Decoder for three errors has been designed and simulated in Xilinx ISE 13.2. From the simulation results, it is observed that the proposed BCH decoder model is area and speed efficient, since the number of slices and delay of the proposed model is lesser than the existing model. Thus area efficiency of 35% is obtained using the proposed BCH decoder structures.

References

1. Xinmiao Zhang and Zhongfeng Wang, “A Low-Complexity Three-Error-Correcting BCH Decoder for Optical Transport Network”, IEEE transactions on circuits and systems—II: express briefs, vol. 59, pp. 10 , Oct 2012.
2. Nahmadi and MH Sirojuddiin, “An Optimal Architecture of BCH Decoder”, Proceedings of IEEE Conference Asian on Information Theory, Indonesia,pp.65-72,Nov-2011 .
3. Hans Kristian, Hernando Wahyono, Kiki Rizki and Trio Adiono, “Ultra-Fast-scalable BCH Decoder with Efficient-Extended Fast Chien Search”, Proceedings of IEEE International Mideast Symposium Circuits and System,china, pp. 5941–5994 , May -2010.
4. Hlou Lamari, EL habti EL idrissi Anas and Elgouri Rachid ,“A Low Power Error Detection in the Chien Search Block for Reed-Solomon Code” ,IEEE Transactions on Information Theory, vol IT-15, pp.122–127 , Feb-2012.
5. R. Huynh, G. Ning, and Y.Huazhung, “A Low Power Error detection in the syndrome calculator Block for Reed- Solomon codes RS: (248, 1881),” Proceedings of IEEE on J. Tsinghua Science and Technology, vol. 14, pp 474-477, August 2009.
6. Hans Kristian, Hernando Wahyono, Kiki Rizki and Trio Adiono, “Ultra-Fast-scalable BCH Decoder with Efficient-Extended Fast Chien Search”, Proceedings of IEEE International Mideast Symposium Circuits and System, pp. 5941–5994 , May -2010.
7. Hung-Yuan Tsai, Chi-Heng Yang, and Hsie-Chia Chang, “An Efficient BCH Decoder with 124-bit Correct ability for Multi-Channel SSD Applications”, Proceedings of IEEE Asian Conferences on Solid-State Circuits,Japan,pp.65-72, November,2012 .