iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** IV **Month of publication:** April 2015

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Complexity Reduction Area Efficient BCH Code for NAND Flash Memory

Ms.C.Gayathri[1], P.Mozhiarasi[2]
[1]*Assistant Professor*,[2]Student (*ME VLSI DESIGN), Electronics and Communication Engineering*
[1,2]*Sri Eshwar College of Engineering,Coimbatore, Tamilnadu*

Abstract-There is a tremendous demand for digital transmission and storage systems due to the increase in development of communication in which data is transferred from transmitter side to receiver side at a faster rate with less time. Even though the time consumption is reduced, there is a chance of occurring errors due to multipath propagation. In order to control these errors Error Correcting Codes (ECC) was invoked, which includes variety of codes depending on application. One of the simplest and first block code is Hamming codes which are capable of correcting only one random error and they are not practically useful. So in this paper, Bose, Chaudhuri and Hocquenghem (BCH) codes form a large class of powerful random error-correcting cyclic codes which can correct multiple errors.Inaddition with BCH, the NAND Flash memory is used to store codeword, and makes for correction part in decoder side where the error is corrected effectively. The enhanced (15, 5, 3) BCH code using hardware description language called VHDL and synthesized in Xilinx 13.2 version.
Key words-Berlekamp Massey Algorithm (BMA), Bose, Chaudhuri and Hocquenghem(BCH), Chein Search (CS), Error Correcting Codes (ECC),Syndrome Calculation(SC)

I. INTRODUCTION

Binary information can be transmitted from one device to another by electric wires or other communication medium. A system that cannot guarantee that the data received by one device are identical to the data transmitted by another device is essentially useless.Yet anytime data are transmitted from source to destination, they can become corrupted in passage.Many factors, including external noise, may change some of the bits from 0 to 1 or vice versa which is illustrated in Fig. 1.

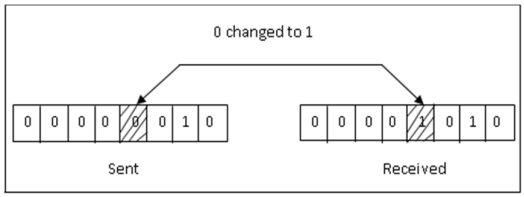


Fig. 1 Error occur during Binary Data Transmission

Binary information/data is transmitted as electromagnetic signal over a channel whenever an electromagnetic signal flows from one point to another; it is subjected to unpredictable interference from heat, magnetism and other forms of electricity. This interference can change the shape or timing of signal.If the signal is carrying encoded binary data, such changes can alter the meaning of data. The purpose of error detection code is to detect such bit reversal errors. Error detection uses the concept of redundancy which means adding extra bits for detecting errors at the destination.

II. MODIFIED TECHNIQUES

A. Nand Flash Memory In Bch Codes

With the continued scaling of NAND flash technology and advancements in multi-level cell technology, flash memory-based storage has gained widespread use in systems ranging from mobile platforms to enterprise servers. Error correcting codes (ECC) are used with NAND flash memory to detect and correct bit errors that may occur with the memory. InNAND flash memory, the bit error rates increases with the number of program/erase cycles and the scaling of technology. The NAND flash memory cell is shown in Fig. 2. Stronger error correcting code (ECC) can be used to support higher rawbit error rates and enhance the lifespan of NAND flash memory.NOR flash technology is fast for reading data but extremely slow on erase and write operation, and NOR flash is more costly than NAND flash. Due to these limitations, most commercial flash products such as solid state

drives (SSD) and various memory cards that are designed today use NAND technology.

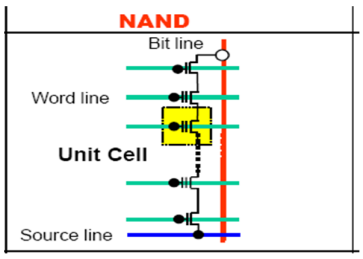


Fig. 2 NAND Flash Memory Cell

B. BCH Codes

In coding theory, the BCH codes form a class of cyclic error correcting codes that are constructed using finite fields.

BCH (Bose ChaudhuriHocquenghem) code is defined by

Codeword length, $n = 2^m-1$

Number of parity check bits, $n\text{-}k \leq mt$

Minimum distance, $d_{min} \geq 2t+1$

where, m– primitive polynomial

k– message length

t – maximum number of correctable errors

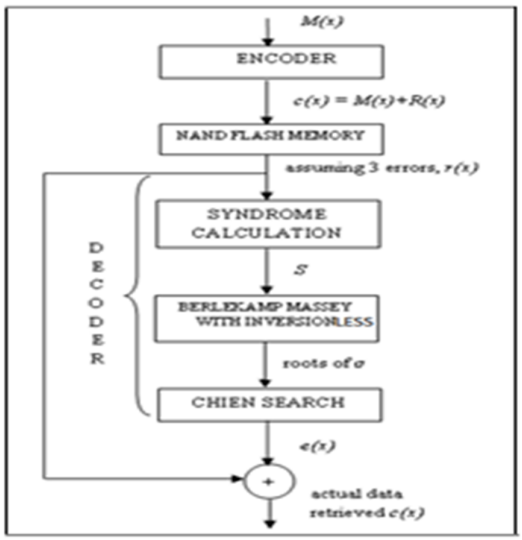
A codeword$(c_0, c_1, ..., c_{n-1})$ of a cyclic code can be represented as the polynomial $c(x) = c_0 + c_1x + ... + c_{n-1}x_{n-1}$.

The generator polynomial of a t-error-correcting BCH code is defined to be the least common multiple (LCM) of $f_1, f_3, ... f_{2*t-1}$, that is,

$g(x) = LCM\{ f_1, f_3, f_5, ... f_{2*t-1}\}$where, f_j is the minimal polynomial of α^j $(0 < j < 2t + 1)$.

In encoder stage, the message bit is padded with zeroes which is then divided with the g(x) to obtain the remainder which is known as redundant bit(R(X)). If remainder is zero then no error in codeword otherwise error (Syndrome) is present. The decoder has three stages namely Syndrome calculation, Berlekamp Massey Algorithm and Chien Search. This syndrome calculation operates over Galois Field (GF) which is also known as Finite Field using TABLE I.

In decoder stage, the BMA is used for finding error locator polynomial. It uses a “Key Equation” by providing known number of coefficients of generating function to determine remaining coefficients of the polynomial. After finding the coefficients of the polynomialthe perfects roots are identified and these roots are givento the CSAlgorithm which finds the errorlocation in the received polynomial by inversing the obtained roots. By performing modulo-2 addition for received polynomial and error pattern polynomial, the actual data is retrieved. The encoding and decoding process is represented in Fig. 3.



Consider $m=4$, $n=2^m-1=2^4-1=15$, $k=5$, $t=3$, $d_{min}=2t+1=2(3)+1=7$. The generator polynomial $g(x)$ is obtained by finding roots of polynomial $P(x)$ of degree m which is primitive if and only if smallest integer n for which $p(x)$ divides x^n+1. Here addition is performed by modulo 2-addition.

$P(x)=(x^{15}+1)$

$=(x+1)\underbrace{(x^4+x+1)}_{p(}\underbrace{(x^{10}+x^9+x^8+x^6+x^5+x^2+1)}_{q(x)}$ (1)

By inversing the root $q(x)$ in (1), generator polynomial g(x) will become $x^{10}+x^8+x^5+x^4+x^2+x+1$.

Let $f_i(x), i=1,2,3,\ldots,2t$ be the minimal polynomial, then $g(x)=LCM\{f_1(x),f_2(x),f_3(x),\ldots,f_{2t}(x)\}$. But, $g(x)$ is simplified to $g(x)=LCM\{f_1(x),f_3(x),f_5(x),\ldots,f_{2t-1}(x)\}$ because every even power of primitive element will have same minimal polynomial as some odd power of the elements having the number of factors in the polynomial.
Therefore, $g(x)=LCM\{f_1(x),f_3(x),f_5(x)\}$where, $f_1(x)=x^4+x+1$, $f_3(x)=x^4+x^3+x^2+x+1$, $f_5(x)=x^2+x+1$.

TABLE I*GF(16)* generated with Primitive polynomial x^4+x+1

Power Form	Polynomial Form	4-Tuple Form $\alpha^3,\alpha^2,\alpha,1$	Decimal Form	Minimal polynomial
0	0	0000	0	x
$1,\alpha^{15}$	1	0001	1	$x+1$
α	α	0010	2	x^4+x+1 $f_1(x)$
α^2	α^2	0100	4	x^4+x+1 $f_2(x)$
α^3	α^3	1000	8	$x^4+x^3+x^2+x+1$ $f_3(x)$
α^4	$\alpha+1$	0011	3	x^4+x+1 $f_4(x)$
α^5	$\alpha^2+\alpha$	0110	6	x^2+x+1 $f_5(x)$
α^6	$\alpha^3+\alpha^2$	1100	12	$x^4+x^3+x^2+x+1$ $f_6(x)$
α^7	$\alpha^3+\alpha+1$	1011	11	x^4+x^3+1
α^8	α^2+1	0101	5	x^4+x+1
α^9	$\alpha^3+\alpha$	1010	10	$x^4+x^3+x^2+x+1$
α^{10}	$\alpha^2+\alpha+1$	0111	7	x^2+x+1
α^{11}	$\alpha^3+\alpha^2+\alpha$	1110	14	x^4+x^3+1
α^{12}	$\alpha^3+\alpha^2+\alpha+1$	1111	15	$x^4+x^3+x^2+x+1$
α^{13}	$\alpha^3+\alpha^2+1$	1101	13	x^4+x^3+1
α^{14}	α^3+1	1001	9	x^4+x^3+1

The minimal polynomial can be obtained by using conjugate roots as mentioned in TABLE II.

TABLE II Minimal Polynomials of the element in *GF(16)*

	Conjugate roots	Minimal polynomial
	0	x
	1	$x+1$
$\varphi_1(x)$	$\alpha,\alpha^2,\alpha^4,\alpha^8$	x^4+x+1
$\varphi_2(x)$	$\alpha^3,\alpha^6,\alpha^9,\alpha^{12}$	$x^4+x^3+x^2+x+1$
$\varphi_3(x)$	α^5,α^{10}	x^2+x+1
$\varphi_4(x)$	$\alpha^7,\alpha^{11},\alpha^{13},\alpha^{14}$	x^4+x^3+1

Those conjugate roots can be found by using $\alpha^{2^i},\alpha^{3^i},\alpha^{5^i},\alpha^{7^i}$ where $i=1,2,3,\ldots$.

1) BCH Encoder

The input/output interface of the re-encoder consists of the following signals:

a) reset (input)- to (synchronously) reset the encoder and synchronise the encoder with the decoder
b) din (input)- the data input
c) dout (output)- the encoded data to be transmitted along the potentially noisy channel
d) clk (input)- the clock signal to clock both the input information and theencoded data
e) vdin (output)- a valid data in signal.

The encoder input/output interface of the re-encoder is shown in Fig. 4

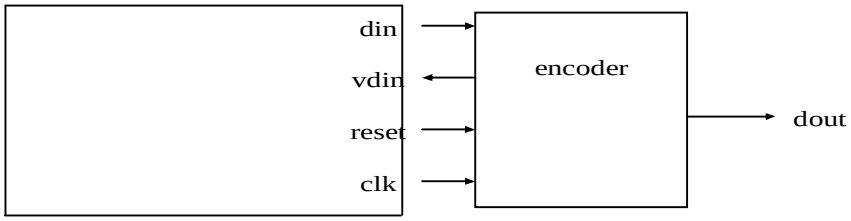


Fig. 4The Input /Output Interface of the Encoder

In initial stage, the message polynomial *(M(x))* is padded with *(n-k)* zero bits to form $X^{n-k}\,M(x)$.

$X^{n-k}\,M(x) = X^{15-5}M(x) = X^{10}\,M(x)$

$= M + 0000000000$ (2)

From (2)$X^{n-k}\,M(x)\ mod\ g(x)$ is computed, the remainder polynomial*(R(x))* is obtained which should be appended with the message polynomial (*M(x)*) to produce the codeword(*c(x)*) i.e., $c(x) = X^{n-k}\,M(x) + R(x)$. This codeword is then stored in flash memory, later for correction purpose in decoder stage. These encoder procedure is illustrated by considering $M(x) = 10001$.

Therefore,

$X^{n-k}M(x) = X^{10}\,M(x) = 10001\,0000000000.$ (3)

Dividing (3) by $g(x)$ w $M(x)$ re $R(x)$ polynomial $R(x) = 1110101100$ which is then appended with $X^{10}\,M(x)$.

$c(x) = 100011110101100$ (4)

Thus, codeword$c(x)$ is obtained finally which is represented in (4).

2) BCH Decoder

The input/output interface of the decoder consists of the following signals:

a) reset (input)- to (synchronously) reset the decoder and synchronise it with the encoder
b) din (input)- the input data
c) dout (output)- the corrected data
d) clk (input)- the clock signal to clock both the received data and the output data
f) vdout (output)- a valid data out signal. A high level on this signal indicates that dout should be clocked with clk.

The input/output interface of the decoder is presented in Fig. 5

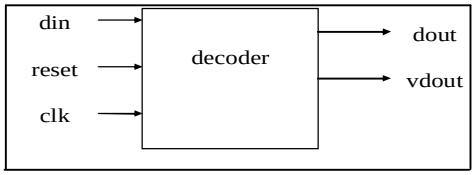


Fig. 5 TheInput/Output Interface of the Decoder

a) Syndrome Calculation (SC)

Let

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

$r(x) = r_0 + r_1x + r_2x^2 + ... + r_{n-1}x^{n-1}$ (5)

$e(x) = e_0 + e_1x + e_2x^2 + ... + e_{n-1}x^{n-1}$ be the transmitted polynomial, the received polynomial and the error polynomial respectively so that$r(x) = c(x) + e(x)$from (5)

$$S_j = \sum_{i=0}^{n-1}(c_i + e_i)\alpha^{i\cdot j} = \sum_{i=0}^{n-1} c_i\alpha^{i\cdot j} + \sum_{i=0}^{n-1} e_i\alpha^{i\cdot j} \quad (6)$$

To generate the syndromes, express (6) as

$S_j = (...((r_{n-1} * \alpha^j + r_{n-2}) * \alpha^j + r_{n-3}) * \alpha^j + ...) * \alpha^j + r_0$.

Thus a circuit calculating the syndrome S_j carries out $(n-1)$ multiplications by the constant value α^j and $(n-1)$ single bit summations in equation (2.12). Note that because $r_j \in GF(2)$ the equation $S_{2i} = S_i^2$ is met.

Let us consider $c(x)$ with 3 errors ($t = 3$) and represent them as $r(x)$ i.e., when noise is added in the channel while transmitting the data to the receiver. At the receiver side, let us assume $r(x) = 1\mathbf{1}001\mathbf{0}110100\mathbf{0}100$. The number of syndrome elements is $2*t = 6$, to find $t=3$ errors. Those syndrome elements are represented as $S_1, S_2, S_3, S_4, S_5, S_6$ in (7).

$$\left.\begin{aligned} S_1(x) &= r(x) \bmod f_1(x) \\ S_2(x) &= r(x) \bmod f_2(x) = r(x) \bmod f_1(x) \\ S_3(x) &= r(x) \bmod f_3(x) \\ S_4(x) &= r(x) \bmod f_4(x) = r(x) \bmod f_1(x) \\ S_5(x) &= r(x) \bmod f_5(x) \\ S_6(x) &= r(x) \bmod f_6(x) = r(x) \bmod f_3(x) \end{aligned}\right\} \quad (7)$$

By the property of Field element several powers of generating element will have the same minimal polynomial (fromTABLE II) when f(x) is polynomial over $GF(2)$, α is an element of$GF(2^m)$. Thus, syndrome polynomial can be represented as

$S(x) = S_1x + S_2x^2 + S_3x^3 + \ldots + S_{2t}x^{2t.}$

Therefore, $S_1(\alpha) = \alpha^{12}$, likewise $S_2(\alpha^2)$, $S_3(\alpha^3)$, $S_4(\alpha^4)$, $S_5(\alpha^5)$, $S_6(\alpha^6)$ syndromes can be calculated.

If the remainder equals zero, then it is declared that no error in the codeword or else error(S) is present. So this error will be a seed for decoding the codeword and to find error location.

b) Berlekamp Massey Algorithm (BMA)

The error location polynomial $\sigma(x)$ is found by $t-1$ recursive iterations. During each iteration r, the degree of $\sigma(x)$ is usually incremented by one. Through this method the degree of $\sigma(x)$ is exactly the number of corrupted bits, as the roots of $\sigma(x)$ are associated with the transmission errors.This method has been reduced by a Berlekamp algorithm and error locator polynomial which is obtained by iterative method as shown in TABLE III.

TABLE III Finding error locator polynomial

μ	$\sigma^{(\mu)}(x)$	d_μ	l_μ	$2\mu-l_\mu$
-1/2	1	1	0	-1
0	1	S_1	0	0
1	$\alpha^{12}x+1$	α^4	1	1
2	$\alpha^7x^2+\alpha^{12}x+1$	α^2	2	2
t=3	$\alpha^{25}x^3+\alpha^5x^2+\alpha^{12}x+1$	-	-	-

Note: 1) Lin and Costello notations were used,

2) d_μ represents discrepancy value

The "Key Equation" is given by

(i)$\sigma^{(\mu+1)}(x) = \sigma^{\mu}(x) + d_\mu d_\rho^{-1} x^{2(\mu-\rho)}\sigma^{(\rho)}(x)$

(ii)$l_{\mu+1} = L = \deg\left(\sigma^{(\mu+1)}(x)\right)$ (8)

(iii)$d_{\mu+1} = S_{2\mu+3} + \sigma_1^{(\mu+1)} S_{2\mu+2} + \sigma_2^{(\mu+1)}$

$S_{2\mu+1} + \cdots + \sigma_L^{(\mu+1)} S_{2\mu+3-L}$

Thus, the Key Equation (8) provides known number of coefficients of generating function to determine remaining coefficients of the polynomial.

c) Chien Search (CS)

The last step in decoding BCH codes is to find the error location numbers. These values are the reciprocals of the roots of $\sigma(x)$ and may be found simply by substituting $1, \alpha, \alpha^2, \ldots, \alpha^{n-1}$ into $\sigma(x)$.
The roots of $\sigma^{(\mu)}(x)$ in $GF(2^4)$ should be found out by following order.

If $\sigma^{(\mu)}(x) = \sigma^{(3)}(x) = \alpha^{25}x^3 + \alpha^5 x^2 + \alpha^{12}x + 1 = 0$ is obtained by substituting $x = 0, 1, \alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6, \alpha^7, \alpha^8, \alpha^9, \alpha^{10}, \alpha^{11}, \alpha^{12}, \alpha^{13}, \alpha^{14}$.Therefore, $\alpha^2, \alpha^6, \alpha^{12}$are the roots for $\sigma^{(3)}(x) = \alpha^{25}x^3 + \alpha^5 x^2 + \alpha^{12}x + 1$. The bit position of error location will be the inverse of their roots $(\alpha^2, \alpha^6, \alpha^{12})$ *i.e., 0***1**000**1**000000**1**000

Thus, the error pattern polynomial can be written as $e(x) = x^{13} + x^9 + x^3$. And actual data is recovered by performing mod-2 addition for $r(x)$ and $e(x)$.

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

$=110010110100100+010001000001000$

$=\underbrace{100011110101100}_{M(x)\qquad R(x)}$ (9)

Therefore, (9) represents the recovered data without any error.

III. SIMULATION RESULTS

The software used to develop programs are Xilinx. The VHDL code for (15,5,3) binary BCH re-encoder and decoder are simulated on Xilinx 13.2i simulator and synthesizer. The simulation waveform for (15,5,3) BCH Re-Encoder, BCH Decoder are shown in Fig. 6 and 7. The waveform simulation has a clock period of 10ns.

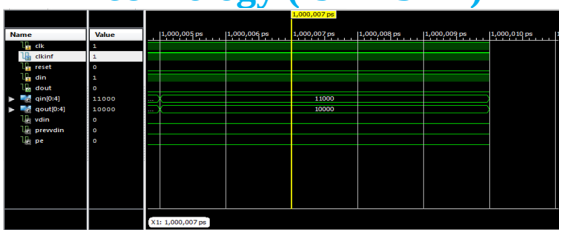


Fig. 6Simulation Waveform of BCH Re-Encoder

In Fig. 6 the clock is set high, which is used to clock the encoded data. The input information data is set to '1'. To synchronize the encoder with decoder the reset signal is used. The input data (din) is set and the encoded output data (dout) is obtained.

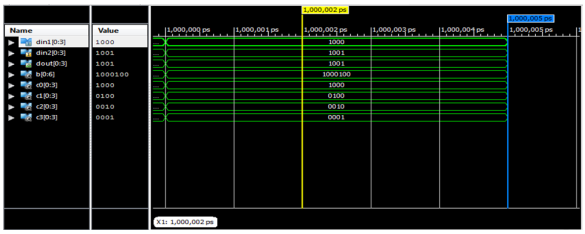


Fig. 7 Simulation Waveform of (15,5,3) BCH Code

In Fig 7, shows the waveform for (15,5,3) BCH code where data input is given and decoded data is obtained using c(i) registers. These c(i) registers are used to store the values of error locator polynomial.

TABLE IV Performance Analysis of (15,5,3) BCH Code

BLOCK	NUMBER OF SLICES	4 INPUT LUT
BCH RE-ENCODER	26	18
SYNDROME CALCULATION	4	7
BERLEKAMP MASSEY ALGORITHM	16	11
CHIEN SEARCH	4	5

From TABLE IV, the performance analysis in terms of slices and LUTs which is done through the simulation.

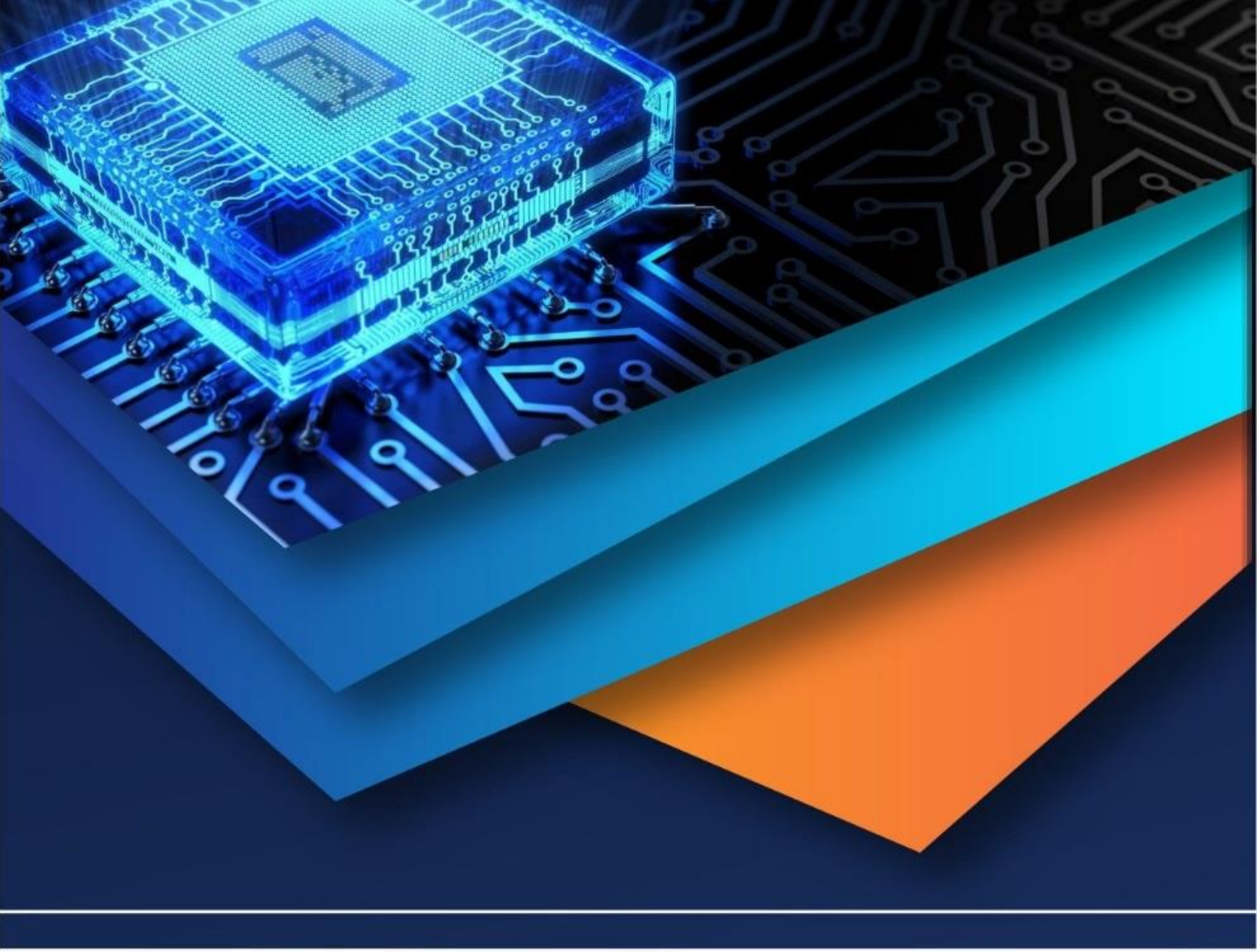
IV. CONCLUSION AND FUTURE WORK

The error correcting technique plays an important role in modern communication system. At first, each character in a text message is transformed to a binary data. (15, 5)BCH encoder and decoder are designed, and it detects and corrects 3 errors by using Berlekamp Massey Algorithm (BMA) and Chien Search (CS) which can be implemented using hardware description language (HDL) known as VHDL and synthesized by Xilinx ISE 13.2 simulator. From the simulation, the performance analysis is done and observed that the enhanced BCH code reduces the area in terms of slices and LUTs which in turn simplify the computational scheduling of the syndrome and Chien search leading to high throughput.

The future work includes: The 15 bit codeword length (n) and 5 bit message length (k) may correct upto three errors (t) while transmission of data. But upto 3 correctable error technique is not enough for real time applications. Due to this reason, the codeword length (n) can be increased along with them a newalgorithm is adopted instead of Berlekamp Massey Algorithm in the next step design.Thus,decoding latency and also complexity can be reduced and can be implemented in Spartan 3 FPGA.

REFERENCES

[1] Arul K. Subbiah and SomnathViswanath (Feb. 2010), "Evolving throughput driven architecture for error correction in NAND memory", Arasan Chip Systems Inc. White Paper.
[2] Berlekamp.E.R (1968), "Algebraic Coding Theory", McGrawHill, New York.
[3] W.W.Peterson, "Encoding and error-correction procedures for the Bose-Chaudhuri Codes", IRE Trans. Inf. Theory, September 1960.
[4] Hyojin Choi, Wei Lin, and Wonyong Sung (May 2010), "VLSI implementation of BCH error correction for multilevel cell NAND flash memory", IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18, no. 5, pp. 843–847.
[5] Hank Wallace (2001), "Error Detection and Correction using BCH Codes".



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY