ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2014

Encoder And Decoder For (15,11,3) and (63,39,4) Binary BCH Code With Multiple Error Correction

R.Elumalai, A.Ramachandran, J.V.Alamelu, Vibha B Raj

Department of Instrumentation Technology, MS Ramaiah of Institute of Technology, Bangalore, India

ABSTRACT: 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. To get the error free communication we need Error correction code. BCH codes invented in 1960s are powerful class of multiple error correction codes with well defined mathematical properties, used to correct multiple random error patterns. The mathematical properties within which BCH codes are defined are the Galois Field or Finite Field Theory. The project proposed is "FPGA implementation of Encoder and decoder for (15, 11, 3) and (63, 39, 4) Binary BCH code using VHDL with multiple error correction". The digital logic implementation of binary encoding and decoding of multiple error correcting BCH code of length n=15 and n=63 over GF (2^4) and GF(2^6)with irreducible primitive polynomials x^4+x+1 and x^6+x+1 are organized into n-k linear feedback shift register circuits for encoding. Iterative decoding algorithms are used to find the location of error and decode the message bits at receiver side. Two encoders and decoders are designed using VHDL to encode and decode the triple and four error correcting BCH code corresponding to the coefficient of generated polynomial. For implementation Spartan 3 FPGA processor is used with VHDL and the simulation & synthesis are performed using Xilinx ISE 13.2.

KEYWORDS: BCH, BCH Encoder, BCH Decoder, FPGA, VHDL, Error Correction, LFSR.

I INTRODUCTION

The Information revolution is in full swing, having matured over the last thirty years. It is estimated that there are hundreds of millions to several billion web pages on computers connected to the Internet, depending on whom you ask and the time of day. The google.com search engine listed 1,346,966,000 web pages in its database. Add to that email, FTP transactions, virtual private networks and other traffic, and the data volume swells to many terabits per day. With all these terabits per second flying around the world on copper cable, optical fiber, and microwave links, the question arises: How reliable are these networks? Imagine losing a newspaper story on the wires every four seconds, or 900 stories per hour. Even a small error rate becomes impractical as data rates increase to stratospheric levels as they have over the last decade. 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. There are many types of error correction codes based on the type of error expected, the communication medium and weather re-transmission, etc. Some of the Error correction codes [3], which are widely used these days, are BCH, Turbo, Reed Solomon, and LDPC. These codes are different from each other in their complexity and implementation [1] [5]. Here, two encoders and decoders are designed using VHDL [6],[8] to encode and decode the triple and quadruple error correcting BCH code corresponding to the coefficient of generated polynomial on FPGA.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2014

The structure of this paper is as follows. Section II contains a brief description of the BCH code and generated polynomial. Section III contains Encoder Design for multiple error correction. Section IV contains decoder Design for multiple error correction. Section V contains simulation results and Section VI contains conclusion and possibility of extending the scope of project.

II LITERATURE SURVEY

The pioneering work lead by Claude Shannon [2] in 1948 with the significant and breakthrough contribution from coding theorist like Hamming, Peterson, Bose-Ray Chaudhari, Hocquenghem, Reed and Solomon, E.Berlekamp, Massey, G.D.Forney Jr. There exists several coding and error correction in digital communication. Bose-Ray Chaudhari [7] presented binary group codes and its error correcting ability. W.W. Peterson [4] describes the encoder techniques for error correction Bose-Ray chaudari binary group codes.

Ernest Jamro [6] used the linear feedback shift register for polynomial division encoder design. The format of the codeword is $c(x) = x^{n-k} * i(x) + b(x)$ and BCH codes are implemented as cyclic code. VHDL implementation and its results have been discussed with reports on recourses used and power consumed.

III BCH CODE

For any positive integers $m(m \geq 3)$ and t $(t < 2^{m-1})$, there exists a binary BCH code with the following parameters:

Block length: $n=2^m-1$
Number of parity check bits: $n-k \leq mt$
Minimum distance: $d_{min} \geq 2t+1$.

The code with the above parameters is capable of correcting any combination of t or fewer errors in a block of n digits with k message bits. The generator polynomial of this code is specified in terms of its roots from the Galois field GF (2^m). The generator polynomial g(X) of the binary t-error-correcting BCH code of length 2^m -1 is given by
g(x)=lcm{Φ1(x), Φ 2(x), Φ 3(x),…….. Φ 2t(x)}. However, generally every even power of α has the same minimal polynomial as some preceding odd power of α in GF(2^4), where α is the primitive polynomial. As a consequence, the generator polynomial for the t-error-correcting binary BCH code can be reduced to **g(x)=lcm{ Φ 1(x), Φ 3(x), Φ 5(x),…….. Φ 2t-1(x)}.** Since the degree of each minimal polynomial is *m* or less, the degree of **g**(X) is at most *mt*. That is, the number of parity-check digits, n-k of the code is atmost equal to *mt.* There is no simple formula for enumerating *n-k*, but if t is small, *n* is exactly equal to *mt.* The *p* parameters for all binary BCH codes of length 2^m-1 with *m<10* are given in Table3. The BCH codes defined above are usually called primitive (or *narrow-sense)* BCH codes.

Let us consider our example of (15,5,3)BCH code. where, n=15, k=5, t=3.
For single error correcting, BCH code of length $n=2^4-1=15$ is generated by $g(x) = \Phi_1(x) = 1+x+x^4$
For double error correcting, BCH code of length n=15 is generated by $g(x) = LCM\{ \Phi_1(x), \Phi_3(x) \} = 1+x^{4+}x^{6+}x^7+x^8$
For triple error correcting, BCH code of length n=15 is generated by $g(x) = LCM\{ \Phi_1(x), \Phi_3(x) \} = 1+x+x^2+x^4+x^5+x^8+x^{10}$
For four error correcting, BCH code of length n=63 is generated by
$g(x)=lcm\{ \Phi 1(x), \Phi 3(x), \Phi 5(x), \Phi 7(x)\}= x^{24}+x^{23}+x^{22}+x^{20}+x^{19}+x^{17}+x^{16}+x^{13}+x^{10}+x^9+x^8+x^6+x^5+ x^4+x^2+x+1$

IV ENCODER LFSR DESIGN

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

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2014

The format of the codeword is as follows :
$c(x) = x^{n-k} * i(x) + b(x)$
Where, codeword $c(x) = c_0 + c_1x + ... + c_{n-1}x^{n-1}$
Information bits $i(x) = i_0 + i_1x + ... + i_{k-1}x^{k-1}$
Remainder $b(x) = b_0 + b_1x + ... + b_{m-1}x^{m-1}$

also, c_i, i_i, b_i are the subsets of Galois field. If b(x) is taken to be the polynomial such that the k data bits will be presented in the codeword, which is given as follows:
$x^{n-k} * i(x) = q(x) * g(x) - b(x)$.

BCH codes are implemented as cyclic code. As a result, the logic which implements encoder_LFSR and decoder is controlled into shift register circuits. With the help of cyclic code properties the remainder b(x) can be calculated in the linear (n-k) stage shift register with the feedback connection to the coefficient of generator polynomial.

The operation of the encoder_LFSR design is as follows :
a) For the clock cycle 1 to k, the original message bits are transmitted without changing its form (during this operation switch s2_in is in position 2), and the linear feedback shift register calculates the parity bits (switch s1_in is on now).
b) For cycle k+1 to n, the generated parity bits in the linear feedback shift register are transmitted (switch s2_in is in position 1) and the feedback in the LFSR is switch off (s1_in off).

The block diagram of BCH encoder_LFSR consists of three modules as shown in fig.1.

- 5 bit Parallel to Serial Shift Register
- Encoder_LFSR module - Linear feedback shift register
- Serial to Parallel Shift Register

In this work, we have designed three error correcting BCH code. The input to the BCH encoder_LFSR is 15- bit message. The BCH
encoder_LFSR uses the linear feedback shift register (LFSR) for polynomial division. This division generates redundant parity bits.

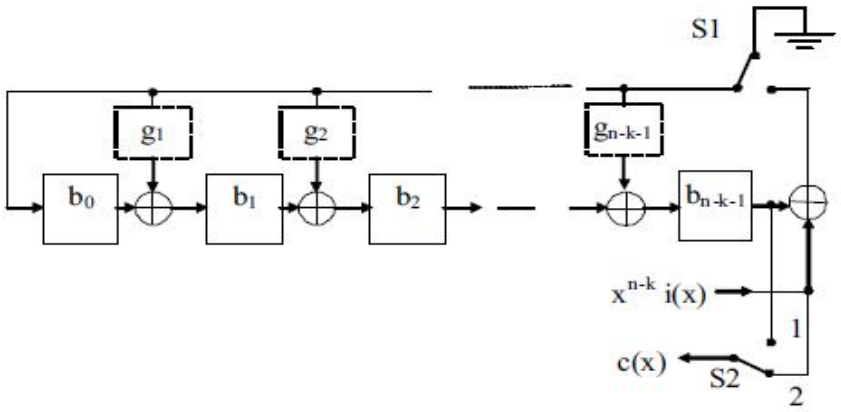

Fig 1 LFSR encoding circuit for (n,k) BCH encoder

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2014

V DECODER DESIGN AND ARCHITECTURE

The BCH decoder has four modules as mentioned below and shown in fig.2:

- Syndrome Calculator
- Solving the key equation
- Error Location
- Error Correction

The implementation and the algorithms used to design the above modules vary with the architectures. The 2nd module, Solving the key equation is the most difficult and complex module as compared to the other modules in respect to the hardware complexity.

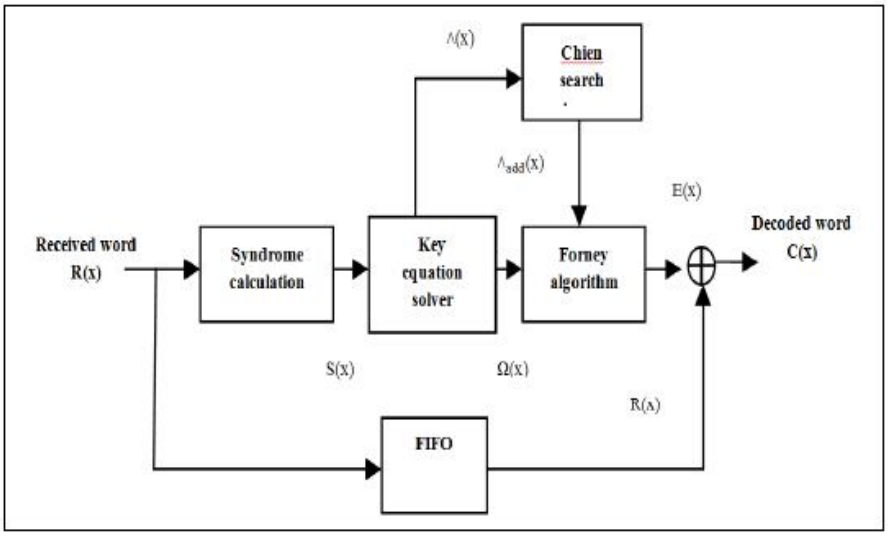

Fig.2 Decoder block diagram

Syndrome Calculator: The syndrome calculator is the first module at the decoder also, the design of this module is almost same for all the BCH code decoder architecture. The input to this module is corrupted codeword. The equations for the codeword, received bits and the error bits are given in equations as follows.

Codeword equation

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

Received bits equation

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

Error bits equation

$e(x) = e_0 + e_1x + e_2x^2 + ... + e_{n-1}x^{n-1}$

Thus, the final transmitted data polynomial equation is given as be

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

The 1st step at the decoding process is to store the transmitted data polynomial in the buffer register and then to calculate the syndromes sj. The important characteristic of the syndromes is that depends on only error location not on transmitted information. The equation of the syndromes are given as follows:

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2014

Define the syndromes S_j as

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

Since $r_j = c_j + e_j$ (j = 0, 1, ..., n-1)
Rewrite the syndrome equation as:

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

By the definition of BCH codes

$$\sum_{i=0}^{n-1} c_i \alpha^{i \cdot j} = 0 \qquad \text{for } (1 \le j \le 2t)$$

Thus,

$$S_j = \sum_{i=0}^{n-1} e_i \alpha^{i \cdot j}$$

The above equation indicates the output of the syndrome calculator. From the equation it can be observed that the syndromes are depends on only error polynomial e(x), so if there is no error occurs during the transmission then all the generated syndromes will be zero.

Key Equation Solver: 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: $\sigma(x) = \sigma_0 + \sigma_1 x + ... + \sigma_t x^t$. The relation between the syndromes and the error location polynomial is given as below:

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

There are various algorithms used to solve the key equation solver. This project is using the Inversion less Berlekamp Massey algorithm to solve the key equation.

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:

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

4. If the value of $\Delta^{(k)}$, 2L ≥ 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
   $T(x) = \Lambda^{(k-1)}(x) / \Delta^{(k)}$
7. Set T(x) = x.T(x).
8. If the value of k< 2t , then go-to step 3
9. Continue for i = 2t – 1 and then End.

The decoder of this project is based on the Inversion-less Berkelamp algorithm (IBM) for Key Equation Calculation. The architecture for iBM algorithm is explained in detail in the next chapter.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2014

Error Location – Chain Search: To calculate the error location is the next step of decoding process, which can be done using chain search block.

Chain Search Algorithm: The roots are calculated as follows

1. For each power of α for ($j = 0$ *to* $n - 1$), α^j is taken as the test root
2. Calculate the polynomial coefficients, of the current root using, coefficients of the past iteration, using, $\Lambda_i^{(j)} = \Lambda_i^{(j-1)} \alpha^i$ during the j^{th} iteration
3. Calculate the sum of the polynomial coefficients

$$\sum_{i=1}^{t} \Lambda_i^j = 1$$

4. The sum is equal to
5. Continue to Step 1 till j = n-1

To locate the roots of the polynomial, chain search block is implemented in hardware for the BCH decoder. The chain search block has a property that the roots going to be a power of α reduce the evolution of the polynomial for every root. So, the use of the chain search block provide the computational benefits of the step, $\Lambda_i^{(j)} = \Lambda_i^{(j-1)} \alpha^i$. This property makes the chain search method more superior then other methods.

Decoder Design Architecture: The Decoder design consists of 5 blocks. These blocks are:

1. Parallel to Serial Shifter
2. Syndrome Block
3. Inversion-less Berlekam Massey Block
4. Chain Search Block
5. Error Correction Block

VI SIMULATION RESULTS

The simulation results of Encoder and decoder is shown in fig.3 & fig.4. The results are simulated with Xilinx ISE13.2

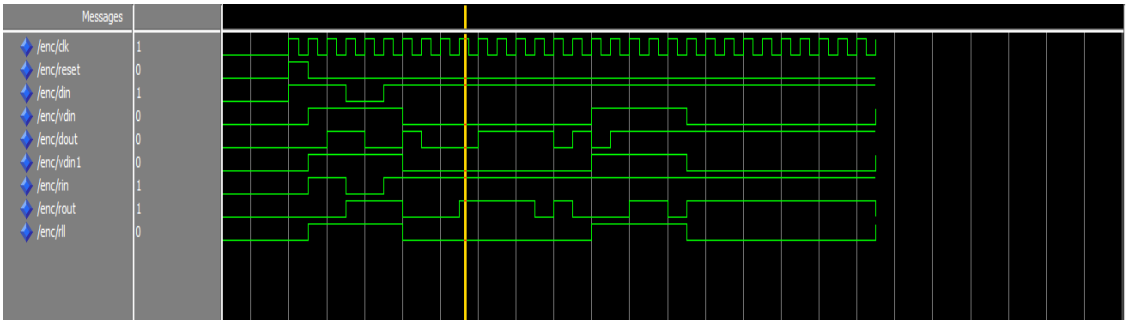

Fig.3 Simulation result for Encoder for 3 bit error

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2014

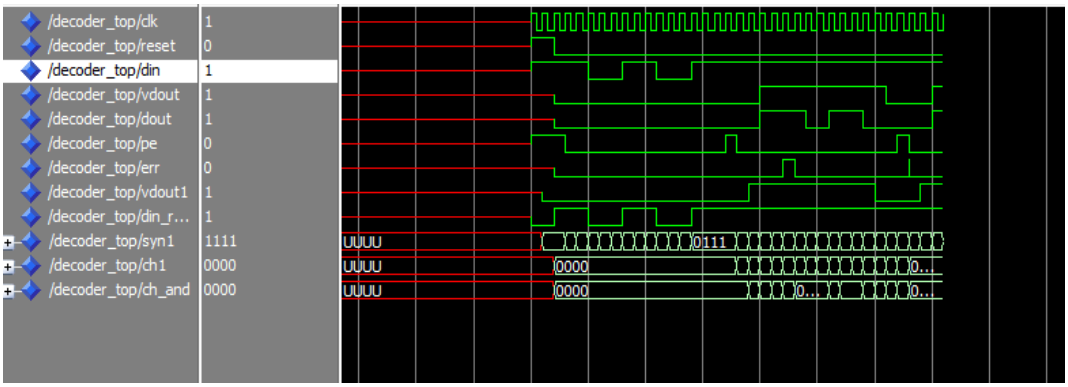

Fig.4 Simulation result for decoder for 3 bit error

VII CONCLUSION

The result presented from the synthesis and timing simulation, shows the (63, 39, 4) BCH Encoder is more advantageous over the other coding techniques according to speed as a parameter. It can correct 4 errors at the receiver side when the original data is corrupted by the noise. When area is considered then (15, 5, 3) is superior and this can correct only 3 bit error. The redundancy is less and the data rate is more. BCH codes have resulted in excellent error correcting codes among codes of short lengths. They are simple to encode and 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.

REFERENCES

[1] M.Y. Rhee - "*Error Correcting Coding Theory*", McGraw-Hill,Singapore, 1989.
[2] C. E. Shannon, ``A mathematical theory of communication," Bell System Technical Journal, vol. 27, pp. 379-423 and 623-656, July and October, 1948.
[3] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*. Upper Saddle River, New Jersey 075458: Prentice Hall, Inc,1995.
[4] W.W. Peterson, "*Encoding and error-correction procedures for the Bose-Chaudhuri Codes*", IRE Trans. Inf. Theory, IT-6, pp. 459-470,September 1960.
[5] Goresky, M. and Klapper, A.M. Fibonacci and Galois representations of feedback-with-carry shift registers, *IEEE Transactions on Information Theory*, Nov 2002, Volume: 48, On page(s): 2826 –2836.
[6] Ernest Jamro, "The Design of a VHDL based synthesis tool for BCH codes", The university of Huddersfiel, September 1997.
[7] R.C. Bose, D.K. Ray-Chaudhuri, "On a class of error correcting binary group codes", Inf. Cntrl, 3, pp. 68-79, March 1960.
[8] H.O. Burton, "Inversionless decoding of binary BCH code", IEEE Trans., 1971, IT- 17, (4), pp. 464-466