International Journal of Engineering Research in Electronic and Communication Engineering (IJERECE) Vol 3, Issue 4, April 2016

# Low Complexity Low Latency Error Correction Using BCH Codes

[1]Pavithra P Vijayan, [2] Rafeekha M J
[1] PG Student [VLSI and ES] [2] Assistant professor,
Department of ECE, TKM Institute of Technology, Kollam
[1]pavithravijayan30@gmail.com,[2] rafeekhauvais2@gmail.com

***Abstract:*** **Bose Choudhuri Hocquenghem (BCH) codes are widely used in applications such as satellite communications, compact disc players, DVDs, disc drives etc. BCH forms a class of error correcting codes that are constructed using finite field. Reed Solomon (RS) codes are used in the previous architecture which forms more complicated finite field operation. In the proposed architecture BCH codes are used in which random error correction is possible. The encoder consists of MPCN (minimal polynomial combination network) block , XOR gates, registers and mux. MPCN blocks are used in architecture which avoids complexity and latency. Decoder consists of syndrome calculator, key equation solver, chien search logic and buffer. The decoder section decodes and corrects the output. Some of the merits of BCH codes are low amount of redundancy, easy to implement in hardware and thus they are widely used. Moreover it is highly efficient in terms of area, speed and power with minimum latency. The VHDL language is used for coding, synthesis was done by using Xilinx ISE and simulated by using ModelSim.**

***Index Terms*— Bose Choudhuri Hocquenghem(BCH), Error Correction Code(ECC) , minimal polynomial combination network..**

## I. INTRODUCTION

Reliability of memory gets affected to a level due to the impact of technology scaling .Major reliability posing concern in many applications are significantly due to the increase in SRAM memory failure. A serious reliability concern in memory applications are both single bit upsets and multiple cell upsets.

In an SRAM cell consist of two n+ nodes in the p-well and two p+ storage nodes in the n-well.In this two sets two potential states that is high state or low state that corresponds to adjacent n+ and p+ nodes. Nuclear spallation reaction takes place between the neutron and the nucleus that is Si of the device, when a ballistic neutron penetrates into the SRAM [6] of the device.The reaction takes place due to the collision of two nuclei with high energy in which the involved nuclei are either disintegrated into the protons, neutrons and light nuclei is called spallation reaction. Nucleons that is the protons and neutrons which are collide with each other in the nucleus of Si, as a result of such a reaction. When they have large value of kinetic energies some of these nucleons escape from the nucleus . This process is called as intra nuclear cascade. Light nuclei gets evaporated from the nucleus of Si, due to this process . Electron-hole pairs along with the ion track is formed when nucleons that is light nuclei and the residual nucleus run inside the SRAM cell. These are called the secondary ions [6].

Mainly through the diffusion process,some of the charges are collected to the storage nodes whenever secondary ions hit the storage nodes and diffusion process occur.Flipping occurs in the logical state of SRAM, then the amount of the charges exceeds a critical charge . Hence soft-error takes place in the SRAM.

In order to meet the varying performance requirements, it is desirable for BCH codec supporting multiple error correcting capabilities, denoted as *t*. However, since the generator polynomials are different for each *t*'s, the hardware cost of encoder is unaffordable if we directly put the logic of each *t*'s together without further simplifications. Nevertheless, because the generator polynomial is defined as the product of several minimal polynomials, and the set of required minimal polynomials of smaller *t* is involved in the set of greater *t*, these minimal polynomials of the greatest *t* can be shared with the one of other smaller *t*'s. Furthermore, by factorizing the generator polynomial into a set of minimal polynomials, the proposed architecture can support arbitrary error correcting capability within the predefined range using the partial set of these minimal polynomials, and the hardware complexity is obviously lower than the conventional linear feedback shift register (LFSR) approach and prior art [8].

Furthermore, to meet the high-throughput, minimal polynomial combination networks (MPCNs) [15] are applied for exploiting higher parallelism. Suppose the minimal polynomial of $\alpha i$ over GF$(2^m)$ be $Mi\ (x) = x^m + M_{i,m-1} x^{m-1} + M_{i,m-2} x^{m-2} + \cdots + M_{i,2} x^2 + M_{i,1} x + 1$, where $M$i, j , $1 \le$ j $\le m - 1$ are binary coefficients of $M$i (x).

## II. LITERATURE REVIEW

Scaling down of semiconductor devices to nano scale which causes short channel, negative bias temperature instability, threshold voltage variation, single bit upsets and multiple cell upsets. Thus to make memory cells as fault tolerant as possible, Error Correction Codes (ECCs) are used. Most commonly used error correction codes are the hamming codes, matrix codes and Reed Solomon codes. Hamming codes

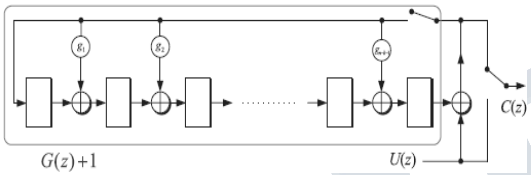


***Fig.1 Conventional LFSR-based BCH encoder***

Are a set of error-correction code in which it is mainly used to detect and correct bit errors. Parity bits concept is used in the hamming code. To ensure the validity of the data these parity bits are added to data so as when it is read or after it has been received in a data transmission. By using more than one parity bit this error correction code can not only identify a single bit error in the data unit, but also its location in the data unit. The parity count in this indicates that single bit errors are detected when the number of ones is incorrect, which in turn reveals that a data bit has been flipped.

Hamming codes which are capable of detecting two bit errors by using more than one parity bit. The number of parity bits required depends on the number of bits that is sent during the data transmission. The hamming code major disadvantage is that it is capable of protecting only those memories that are affected by single bit upsets. Reed–Solomon (RS) codes, an architecture combining the encoder and syndrome calculator is proposed in [10]. RS codes presents a method of factorization of the generator polynomial into the product of $(x - \alpha i)$. However, if the same method is applied to BCH code, the generator polynomial, which consists of minimal polynomials that is Mi (x) with binary coefficients, will be factorized into the product of $(x - \alpha i)$, and the encoding process will be transformed from binary operation into more complicated finite field operation. Therefore, it is not efficient for BCH codes.

For BCH codes, a combined encoder and syndrome calculator (SC) is proposed in [9]. By preprocessing the received codeword with encoder, the evaluation range of SC is reduced from (n − 1) degree to (n − k − 1) degree. Nevertheless, the preprocessing of received codeword increases the total latency of SC, and the hardware resources cannot be shared with this method. In contrast, because the syndrome coefficients Si can be calculated with minimal polynomials [17], and the proposed encoder architecture is based on minimal polynomials as well, these two major components of BCH codec can be combined together efficiently.

## III. PROPOSED ARCHITECTURE

### *A. BCH code*

Data communication is used in the transfer of data and maintenance of the data during the process. Exchange of data takes place between a source and a receiver. Errors may be introduced during transmission from the source to a receiver, and therefore many communication channels are subject to channel noise. Secure communication is very important, not only to protect the confidentiality of the data but also to retain the correctness of the data. Secure communication can be implemented by various methods. Every method has its own advantages and disadvantages. BCH code is used in this to avoid these soft errors. Since this codes can correct a large number of errors. LFSR based BCH encoder is used to reduce this MPCN based BCH codec architecture is used. By using this complexity reduced decoded output can be obtained. In the BCH code finite field operations are used. The definitions and main results underlying finite field theory are presented .

For a finite field GF$(2^m)$ it involves $2^m$ elements and can be viewed as a m dimensional vector space over the subfield GF(2). Suppose β be an element in GF$(2^m)$. However, β can also be a root of another polynomial over GF(2) with degree less than $2^m$. The minimal polynomial of β is defined as the monic polynomial with the smallest degree among all polynomials over GF(2) having β as a root. In a monic polynomial, the coefficient of the highest order is always 1. If there were two minimal polynomials, the difference will have a smaller degree and will still have β as a root. Since the above result contradicts its basic definition, the minimal polynomial is unique. Besides, a minimal polynomial must be irreducible over GF(2); otherwise, it will

not be the least degree polynomial having β as a root. The roots of minimal polynomial are conjugates of each other.

### B. Finite fields

Finite fields rely to a large extent on powerful and elegant algebraic structures called error control codes. In a field it contains a set of elements in which it is possible to add, subtract, multiply and divide field elements and always obtain another element within the set. A field containing a finite number of elements is called a finite field.

### C. Field definitions and basic feature

The concept of a field is now formally introduced. A field is a non-empty set of elements with two operators called addition and multiplication, with + and * is denoted respectively. For F to be a field a number of conditions must hold. The following are some of the conditions for a field used in this paper.

Closure: For every a, b in F
c = a + b; d = a * b;
where c, d ϵ F.
Associative: For every a, b, c in F
a + (b + c) = (a + b) + c and a * (b * c) = (a * b) * c
Identity: There exists an identity element 0 for addition and 1 for multiplication satisfy 0 + a = a + 0 = a and a * 1 = 1 * a = a for every a in F.
Inverse: If a is in F, there exist elements b and c in F such that a + b = 0 a * c = 1 Element b is called the additive inverse, b = (-a), element c is called the multiplicative inverse, c = a+1 (a not equal to 0)
Commutative: For every a, b in F
a + b = b + a a * b = b * a
Distributive: For every a, b, c in F
(a + b) * c = a * c + b * c

### D. Construction of BCH code

BCH codes can be defined by the two parameters that are code size of n and the number of errors to be corrected is t.
Block length , $n = 2^m - 1$
Number of information bits : $k \geq n\text{-}m*t$
Minimum distance: $d_{min} \geq 2t + 1$
Let $m_i(x)$ be the minimum polynomials of $\alpha^i$ then generator polynomial G(x) can be computed from the minimal polynomial.

***Table.1 The elements of GF ($2^4$) generated by p(x) =1+x+$x^4$***

| Field of 16 elements generated by $x^4 + x + 1$ | | |
|---|---|---|
| Power Form | n-Tuple Form | Polynomial Form |
| 0 | 0000 | 0 |
| 1 | 0001 | 1 |
| $\alpha$ | 0010 | $\alpha$ |
| $\alpha^2$ | 0100 | $\alpha^2$ |
| $\alpha^3$ | 1000 | $\alpha^3$ |
| $\alpha^4$ | 0011 | $\alpha + 1$ |
| $\alpha^5$ | 0110 | $\alpha^2 + \alpha$ |
| $\alpha^6$ | 1100 | $\alpha^3 + \alpha^2$ |
| $\alpha^7$ | 1011 | $\alpha^3 + \alpha + 1$ |
| $\alpha^8$ | 0101 | $\alpha^2 + 1$ |
| $\alpha^9$ | 1010 | $\alpha^3 + \alpha$ |
| $\alpha^{10}$ | 0111 | $\alpha^2 + \alpha + 1$ |
| $\alpha^{11}$ | 1110 | $\alpha^3 + \alpha^2 + \alpha$ |
| $\alpha^{12}$ | 1111 | $\alpha^3 + \alpha^2 + \alpha + 1$ |
| $\alpha^{13}$ | 1101 | $\alpha^3 + \alpha^2 + 1$ |
| $\alpha^{14}$ | 1001 | $\alpha^3 + 1$ |

$$G(x) = LCM [m_1(x), m_3(x), \ldots., m_{2t}(x)] \quad (1)$$

In this work n=15, k=7 and for this t=2 is considered. Hence the generator Polynomial with,a, a2,... a4 which are roots that are obtained by multiplying the following minimal polynomials:

$$m1(x) = 1+x+x4 \quad (2)$$

$$m3(x) = 1+x+x2+x3+x4 \quad (3)$$

Substituting m1(x) and m3(x) in equation generator polynomial is obtained.

$$G(x) = LCM \{m1(x), m3(x)\} \quad (4)$$

$$G(x) = \{(1+x+x4) (1+x+x2+x3+x4)\} \quad (5)$$

$$G(x) = 1+x4+x6+x7+x8 \quad (6)$$

To build BCH codes over galois field, we need to find out elements of galois field generated by p(x) =1+x+x4 is given in table above.

### E. MPCN Based BCH Encoder

In the minimal polynomial combinational network (MPCN) based BCH encoder instead of linear feedback shift register(LFSR) MPCN blocks are used to avoid hardware complexity.

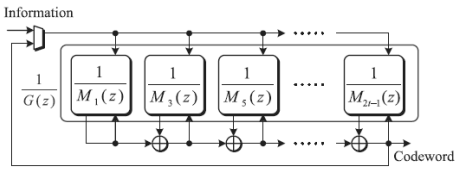


*Fig.2 Block diagram of BCH encoder*

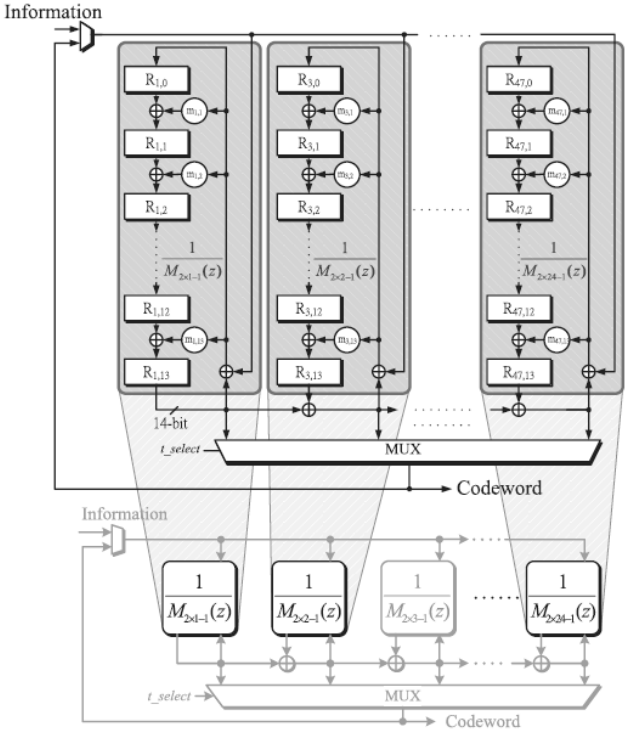


*Fig.3 Block diagram of BCH encoder*

In this multiplication operation of the inputs is done with registers to obtain the codeword. The information bits are xored with the generator polynomial. Generator polynomial is calculated by the galois field. In the galois field arithmetic operation such as addition,subtraction and multiplication is possible.

In the block diagram each block consists of several minimal polynomials. Generator polynomials are formed from the several minimal polynomials. The codeword is formed by information polynomial by generator polynomial. To remove the path that causing a zero-delay loop in, it can be modified by adding input to each block. The data bits stored in registers at the (k+1)th cycle are not the parity bits, which are continuously computed using the feedback signal in the last (nk) cycles in the linear feedback shift register.

The figure consists of multiplexers, registers and minimal polynomial block and xor gates. Information is xored with the generator polynomial and the resultant value is stored in the register. All the output are xored and given to the multiplexer. Thus after operation the resultant is appended with the information bits to obtain the codeword. In the comparison between our proposed encoder architecture and conventional LFSR architecture is shown , two different groups of error correcting capabilities are provided .

### *F. BCH Decoder*

In the BCH decoder architecture consists of syndrome calculator,key equation solver,chien search logic and buffer.The figure below shows the BCH decoder. Different types of error correcting circuit are there which can correct single bit, double bits and multiple bits.

The above figure is the detailed diagram of the BCH decoder which can correct multiple errors. It consists of syndrome calculator, syndrome re arranger, BMA, chien search unit, buffer and control unit. First the received bits are stored in the buffer. To the syndrome calculator unit syndrome bits are

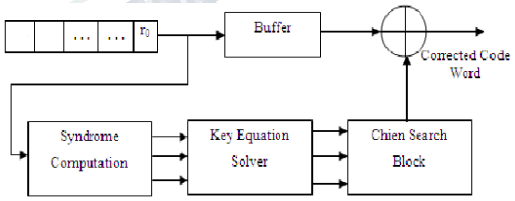


*Fig.4 BCH decoder*

Given and in the syndrome re arranger module syndrome bits from the syndrome calculator are rearranged. The output of the syndrome rearranger is given to the Berlekamp Multiplier Accelerator(BMA) unit.BMA unit consists of summation block , multiplication block and AND gates. Polynomial binary value is taken and multiplied then it is given to the addition block. Sum of product obtained from the BMA block is given to the chien search unit.The chien search unit calculates the position in the error of the received codeword that is stored in the buffer. Then the corrected output of the received codeword is obtained at the output. The buffer is used to store the value.

The decoding process for BCH code consists of four major steps

- Syndrome computation
- determine coefficient of error locator polynomial
- to find the root of error locator polynomial
- and finally error correction in the bits is done

### G. Calculation of the syndromes

Let

$$c(x) = c0 + c1x + c2x2 + ::: + cn-1xn-1 \quad (7)$$

$$r(x) = r0 + r1x + r2x2 + ::: + rn-1xn-1 \quad (8)$$

$$e(x) = e0 + e1x + e2x2 + ::: + en-1xn-1 \quad (9)$$

Be the transmitted polynomial which is c(x), the received polynomial that is r(x) and the error polynomial can be e(x) which can be represented as by the following equation received codeword = transmitted polynomial+ error polynomial.The first step of the decoding process is storing the received polynomial r(x) in a buffer register and to calculate the syndromes Sj . The most important feature of the syndromes is that they do not depend on transmitted information but only on error locations

### H. Solving the key equation

In the second stage of the decoding process in which the coefficients of the error location polynomial using the syndromes is found . The relationship between the syndromes and the error location polynomial is given by an equation and the roots of error location polynomial error positions. Coefficients of the error location polynomial are calculated by the Peterson-Gorenstein-Zieler algorithm or Euclids algorithm . In this thesis the Berlekamp-Massey Algorithm (BMA) has been used as it IS the most efficient method in practice.

### I. Finding the error locations

The error location is the last step in decoding process of BCH codes. These values are the reciprocals of the roots of error location polynomial.

## IV. RESULTS AND DISCUSIONS

The modules are modelled using the VHDL in Xilinx ISE Design Suite 8.1i and the simulation of the design is performed using the Modelsim SE 6.3f to verify the functionality of the design. Here using BCH code with minimal polynomial combinational network which can correct upto three bit error is implemented.

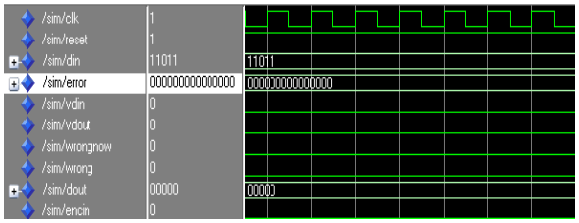


*Fig.5 Simulation of input when reset = 1*

When input = 11011
Reset = 1

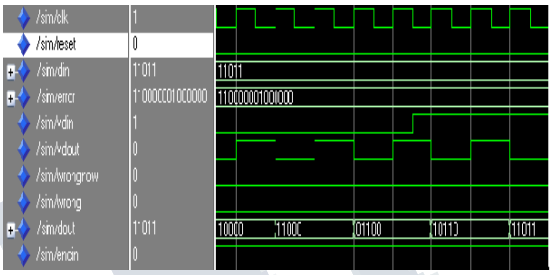


Fig.6 Simulation of the output
When input = 11011
Reset = 0
Output = 11011

## V. CONCLUSION

Soft errors caused due to multiple cell upsets can lead to serious problems in memory applications. Thus to avoid such errors Bose-Chaudhuri-Hocquenghem (BCH) code is used in which random errors can be corrected .The use of this code maximizes the error detection and correction capability. But the main disadvantage of this is complexity and latency. Moreover BCH based decoder algoritham is more complex and consume more time. So in order to avoid this Minimal Polynomial Combinational Network (MPCN) based BCH encoder and decoder are implemented. Thus complexioty and area can be reduced and it only consumes less power. The whole architecture is designed using VHDL, synthesized using Xilinx ISE8.1i and simulated using ModelSim SE 6.3f simulator.As an extention of this work, codec with better encoding and decoding which consumes less area and power is intended to perform as future work.

## REFERENCES

[1] Chi-Heng,Yi-Min Lin,Hsia-Chia Chang and Chen-Yi Lee,"An MPCN Based BCH Codec ArchitectureWith Arbitrary Error Correcting Capability", IEEE TrancationsOn Very Large Scale Integration Systems.VOL.23,NO.7,JULY 2015.

[2] K. Lee, S. Lim, and J. Kim,"Low-cost, low-power and high-throughput BCH decoder for NAND flash memory,", in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS),May 2012, pp. 413415.

[3] Y. Cai, E. F. Haratsch, O. Mutlu, and K. Mai, "Error patterns in MLC NAND flash memory: Measurement, characterization, and analysis", in Proc. Design, Autom.Test Eur. Conf. Exhibit. (DATE), Mar. 2012,pp. 521526.

[4] W. Liu, J. Rho, and W. Sung,"Low-power high-throughput BCH error correction VLSI design for multi-level cell NAND flash memories, ", in Proc. IEEE Workshop Signal Process. Syst. Design Implement., Oct. 2006, pp. 303308.

[5] T.-H. Chen, Y.-Y. Hsiao, Y.-T. Hsing, and C.-W.Wu,"An adaptive-rate error correction scheme for NAND flash memory", in Proc. 27th IEEE VLSI Test Symp. (VTS),May 2009, pp. 5358.

[6] S. Li and T. Zhang,"Improving multi-level NAND flash memory storage reliability using concatenated BCH-TCM coding,", IEEE Trans.Very Large Scale Integr.(VLSI) Syst., vol. 18, no. 10, pp. 14121420,Oct. 2010.

[7] S. Tanakamaru, C. Hung, and K. Takeuchi,"Highly reliable and low power SSD using asymmetric coding and stripe bitline-pattern elimination programming, ", IEEEJ. Solid-State Circuits, vol. 47, no. 1, pp. 8596, Jan. 2012.

[8] Y.Lee, H. Yoo, I. Yoo, and I.-C. Park,"6.4 Gb/s multi-threaded BCH encoder and decoder for multi- R. Cherukuri, "Agile encoder architectures for strength-adaptive long BCH codes",in Proc. IEEE GLOBECOM Workshops, Dec. 2010,pp. 19001904.

[9] Y.M. Lin, C.-H. Yang, C.-H. Hsu, H.-C. Chang, and C.-Y. Lee "A MPCN-based parallel architecture in BCH decoders for NAND flash memory devices", IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 58,no. 10, pp. 682686, Oct. 2011.

[10] G. Fettweis and M. Hassner, "A combined Reed-Solomon encoder and syndrome generator with small hardware complexity", in Proc. IEEE Int. Symp. Circuits Syst.(ISCAS), vol. 4. May 1992, pp. 18711874.

[11] S. Lin and D. J. Costello, *Error Control Coding*,2ned. Englewood Cliffs, NJ, USA: Prentice-Hall, 1983.