Journal of Electronics and Informatics (2020)
Vol.02/ No.04
Pages: 220-227
https://www.irojournals.com/iroei/
DOI: https://doi.org/10.36548/jei.2020.4.004

BCH Encoder and Decoder for Emerging Memories

Rohith R[1], Saji A J[2]
[1,2]Department of Electronics and Communication Engineering, Rajiv Gandhi Institute of Technology, Kottayam, India.
Email: [1]rohithanayadi@gmail.com, [2]aj.saji@gmail.com

Abstract- In this paper, an encoder and decoder system is proposed using Bose-Chaudhuri-Hocquenghem (BCH) double-error-correcting and triple-error detecting (DEC-TED) with emerging memories of low power and high decoding efficiency. An adaptive error correction technique and an invalid transition inhibition technique is enforced to the decoder. This is to improve the decoding efficiency and reduce the power consumption and delay. The adaptive error correction gives high decoding efficiency and invalid transition technique reduce the power consumption issue in conventional BCH decoders. The DEC-TED BCH decoder combines these two techniques by using a specific Error Correcting Code Clock and Flip Flops. This technique provides an error correcting encoder and decoder solution for low power and high-performance application using emerging memories. The design simulated in Xilinx FPGA using ISE Design Suite 14.5.

Keywords: Bose-Chaudhuri-Hocquenghem (BCH) code; Adaptive error correction; Invalid transition inhibition; emerging memories

I. INTRODUCTION

Emerging Memories are brand new ideas, vying for the position of mainstream replacement [5]. Emerging memories includes phase change memory, resistive random access memory (ReRAM), phase change RAM (PRAM) and spin-transfer torque magneto resistive random access memory (STT-MRAM). Emerging memories have faster write cycle, faster access time and endurance than ordinary memory. Their performance and density are better than DRAM (Dynamic Random Access Memory), NOR and NAND flash memory, so that they are called storage class memories. They are flexible, non-volatile and have efficient memory hierarchy, low-latency characteristics and high-density. Besides SCMs (Single Chip Memories), some emerging memories, namely STT-MRAM, are perfect replacement for embedded memories because of their low latencies and low power consumption. The emerging memories are fighting with diminished reliability, when the memory scales down, as a solution for this problem error-correcting code and its encoder or decoder circuits have developed. NAND and NOR flash require a powerful Error Correcting Codes (ECC) in which maximum of 100 errors can be corrected. With the help of an ECC, that can correct two or three errors, the emerging memories can reach the required memory yield. Error Correcting Codes are used to enhance the performance of the memory regarding to density and power consumption and thereby increase the memory yield. So, ECC has become an important element of emerging memories [1]. For emerging memories, Bose Chaudhuri Hocquenghem (BCH)are used to correct two or three errors. Their standard monotonous and sequential decoding processes require 2(n+t) cycles, where t is the number of errors that can correct and n is the length of the code. This nature of BCH codes are not suitable with emerging memories. This is because, of the emerging memory with a latency of few nanosecond. A fully parallel BCH decoder structure with combinatorial logic gates has developed, that can correct two error with a latency of few nano second and are called Double Error Correcting (DEC) BCH decoder. In this paper, we introduce a BCH Encoder and a Decoder. The decoder has low-power consumption and high decoding efficiency. The BCH decoder is DEC and has triple error detecting capacity suitable for emerging memories. In ordinary BCH decoders, to reduce power consumption and average delay, an adaptive error correction technique is used in DECTED BCH decoder. To reduce the power consumption, an invalid transition technique including a ECC clock and flip flops are used. Introducing a DEC-TED BCH encoder and decoder of 84-bit data.

The remaining section of this paper is organized as follows. In section II, an over view of BCH codes and Error correction. In III, the proposed encoder and decoder. Section IV introduces the synthesis and comparison results of the proposed DEC-TED decoders and DEC-TED existing decoder [1]. Finally, section V concludes this paper.

Table 1
Relationship Between the Number of Errors and Syndrome Vector

Number of Errors	s_0	s_1 and s_3
Non	0	$s_1 = s_3 = 0$

ISSN: 2582-3825 (online)
Submitted: 22.12.2020
Accepted: 12.01.2021
Published: 19.01.2021

Journal of Electronics and Informatics (2020)
Vol.02/ No.04
Pages: 220-227
https://www.irojournals.com/iroei/
DOI: https://doi.org/10.36548/jei.2020.4.004

Single-bit	1	$s_1^3 = s_3$
Double-bit	0	$s_1^3 \neq s_3$
Triple-bit	1	$s_1^3 \neq s_3$

II. BCH CODES AND ERROR CORRECTION

A. Primitive Binary BCH Codes

These codes are the part of cyclic codes. For achieving good error correction, the roots of the generator polynomials are specified carefully. Each BCH code is t error correcting code in that it can detect and correct up to *t* random error per code word. A binary BCH code is defined over a finite called Galois field (2^m) with 2^m elements, where m is the degree. The (n, k, d) BCH code over GF(2^m) is represented as follows:[1]

Block length: $n = 2^m - 1$
Number of message bits k: n - mt
Minimum distance: $dmin \geq 2t + 1$

The decoding process for the DEC-TED BCH are described below.

1) Calculation of syndrome vector:

For (n, k, d) DEC-TED BCH code, the parity-check matrix H is given by

$$H = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \alpha & \alpha^2 & \dots & \alpha^{n-1} \\ 1 & \alpha & \alpha^3 & \dots & \alpha^{3(n-1)} \end{bmatrix} = \begin{bmatrix} 1 \\ H_1 \\ H_3 \end{bmatrix}$$

To detect the presence of error in the received code word y, syndrome vector s is calculated. Non zero syndrome indicates the presence of error in the received code word.:

$$s = y.H^T = [y.1^T \quad y.H_1^T \quad y.H_3^T] = [s_0 \quad s_1 \quad s_2]$$

S_0 is a 1-bit vector, and S_1 and S_3 are m-bit vectors for the code generated in GF(2^m):

2) Finding the Error Location Polynomial:

The second decoding step is to find the error location polynomial (ELP) [4] related to the calculated syndrome vectors. For a DEC-TED BCH code, the error location polynomial can be represented as:

$$\sigma(x) = 1 + \sigma_1(x) + \sigma_2(x^2) \quad (1)$$

If the codewords are generated on GF(2m) each coefficient of the error location polynomial is an m-bit vector. The Berlekamp Massey (BM) algorithm is used to calculate the coefficients of the ELP.

3) Spotting the Error Locations:

After getting the coefficients (σ_1 and σ_2), the Chien search algorithm is applied to detect the roots of the ELP by substituting n elements of GF(2^m), $\alpha^0, \alpha^1, \dots, \alpha^{n-1}$ into (2.1).

4) Error Correction:

From the step3 , an error vector, v, is acquired, and a corrected codeword, y^* , can be represented as $y^* = y + v$. This can be implemented using XOR gates.

B. Error Correction

Conventional codes are generally used as both channel coding techniques and as low density parity check codes and building blocks in turbo codes. For each n input bits, the encoder produces m output bits where $m \geq n$ *((m,n)* code). A parity check matrix *H* of dimension *(m - n) * n* is used to check whether the input code word is valid [2]. The Error Correction Codes scheme can be applied to the parallel filter considered by defining a set of check filters [2].

$$Z_1[n] = y_1[n] + y_2[n] + y_3[n]$$
$$Z_2[n] = y_1[n] + y_2[n] + y_4[n] \quad (2)$$
$$Z_3[n] = y_1[n] + y_3[n] + y_4[n]$$

ISSN: 2582-3825 (online)
Submitted: 22.12.2020
Accepted: 12.01.2021
Published: 19.01.2021

Journal of Electronics and Informatics (2020)
Vol.02/ No.04
Pages: 220-227
https://www.irojournals.com/iroei/
DOI: https://doi.org/10.36548/jei.2020.4.004

x[n] is the input data and *y[n]* is the output data. An error on y_1 will insert error on the checks of Z_1, Z_2, Z_3. Likely errors on the other bit causes errors on different z_i. Therefore, error can be spotted and corrected using the conventional error correcting codes. Using the remaining data and check outputs the error is corrected by reconstructing erroneous bit. For example, when an error on y_3 is detected, it can be corrected by making

$y_{c3}[n] = z_1[n] - y_2[n] - y_3[n]$.

III. PROPOSED SYSTEM

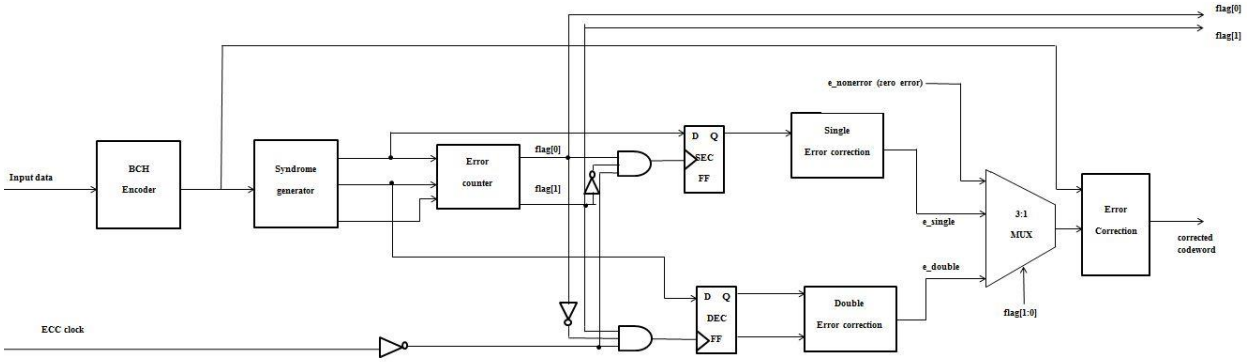

Figure 1 Block diagram of DEC-TED BCH Encoder and Decoder

Figure1 shows the block diagram of DEC-TED BCH encoder and Decoder.

A. *BCH Encoder*

A DEC-TED Encoder is identical to DEC Encoder. The difference is that that an additional parity bit is added to the encoder [6]. Since this is an overall parity bit covering the data as well as check bits. In the *(n, k)* block code, *n* represents the codeword and *k* represents the original information bits. *(n - k)* number of parity bits are added to the original message bits. This BCH code is a part of block codes. In a block code, codeword is the mixture of an information bits and parity bits.

The bits which hold messages are information bits, and there are also parity bits for security. The parity bits also guaranteed that the codeword has correct structure, needed for block codes. Encoder generate the parity bits and train them to the original information bits. For the generated codeword n the information bit is of *k* and parity bits are of *r* gives $n = k + r$.

In this work we carried out (15, 7) BCH encoder. Bose- Chaudhuri-Hocquenghem codes whose generated polynomial have the roots specified over the Galois Field GF(2^m). α be the primitive element in GF(2^m). The generator polynomial *g(i)* of the code is the lowest degree polynomial *(g(i))* over GF(2). Let *m(i)* be the minimum polynomial of α_i then generator polynomial *G(i)* can be computed

$G(x) = LCM [m_1(i), m_3(i), …., m_4(i)]$

In this paper n = 15, k = 7 and t = 2 is taken and the generator polynomial whose roots $\alpha_1, \alpha_2, ..., \alpha_4$ are obtained by the multiplication of the following minimal polynomials:

$m_1(i) = 1 + i + i^2$
$m_3(i) = 1 + i + i^2 + i^3 + i^4$

Substituting $m_1(i)$ and $m_3(i)$ in the equation generator polynomial is obtained.
$G(i) = LCM [m_1(i), m_3(i)]$
$G(i) = (1 + i + i^2) (1 + i + i^2 + i^3 + i^4)$
$G(i) = 1 + i^4 + i^6 + i^7 + i^8$

The parity bits P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9, P_{10} are computed as a function of the input message bits i_1, i_2, i_3, i_4, i_5, i_6, i_7, i_8, i_9, i_{10}, i_{11}, i_{12} as follows

ISSN: 2582-3825 (online)
Submitted: 22.12.2020
Accepted: 12.01.2021
Published: 19.01.2021

Journal of Electronics and Informatics (2020)
Vol.02/ No.04
Pages: 220-227
https://www.irojournals.com/iroei/
DOI: https://doi.org/10.36548/jei.2020.4.004

$P_0 = i_1 + i_2 + i_3 + i_4 + i_5 + i_6 + i_7 + i_8 + i_9 + i_{10} + i_{11}$
$P_1 = i_1 + i_2 + i_3 + i_4 + i_5 + i_6 + i_7 + i_8 + i_9 + i_{10} + i_{12}$
$P_2 = i_1 + i_2 + i_3 + i_4 + i_5 + i_6 + i_7 + i_8 + i_9 + i_{12} + i_{11}$
$P_3 = i_1 + i_2 + i_3 + i_4 + i_5 + i_6 + i_7 + i_8 + i_{12} + i_{10} + i_{11}$
$P_4 = i_1 + i_2 + i_3 + i_4 + i_5 + i_6 + i_7 + i_{12} + i_9 + i_{10} + i_{11}$
$P_5 = i_1 + i_2 + i_3 + i_4 + i_5 + i_6 + i_{12} + i_8 + i_9 + i_{10} + i_{11}$
$P_6 = i_1 + i_2 + i_3 + i_4 + i_5 + i_{12} + i_7 + i_8 + i_9 + i_{10} + i_{11}$
$P_7 = i_1 + i_2 + i_3 + i_4 + i_{12} + i_6 + i_7 + i_8 + i_9 + i_{10} + i_{11}$
$P_8 = i_1 + i_2 + i_3 + i_{12} + i_5 + i_6 + i_7 + i_8 + i_9 + i_{10} + i_{11}$
$P_9 = i_1 + i_2 + i_{12} + i_4 + i_5 + i_6 + i_7 + i_8 + i_9 + i_{10} + i_{11}$
$P_{10} = i_1 + i_{12} + i_3 + i_4 + i_5 + i_6 + i_7 + i_8 + i_9 + i_{10} + i_{11}$

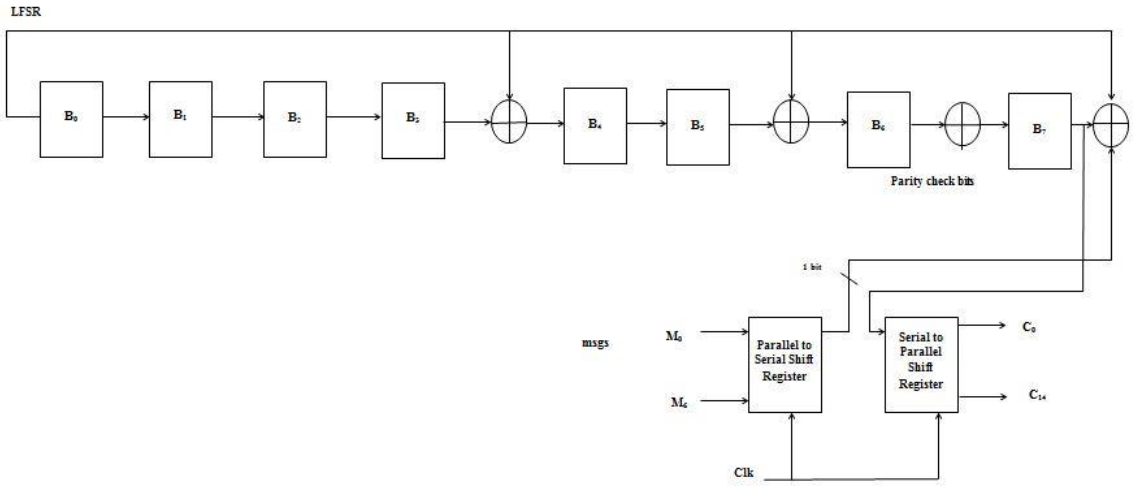

Figure 2 Block diagram (15,7) BCH Encoder

Figure 2 shows the block diagram of (15, 7) BCH Encoder. In this Paper we are using 84-bit data and split the data into 12 blocks, each of 7 bit and is given to the (15, 7) BCH encoder.
The encoding is done by y = x ● G where, y is the output, x is the input and G is the generator matrix.
The ● operation is based on the multiplication and XOR (modulo two addition) [3].

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

B. DEC-TED BCH Decoder with Adaptive Error Correction

The proposed adaptive error correction techniques is shown in Figure 3. The first block is the syndrome generator. The syndrome is computed by s = y ● H^T, where H is the parity check matrix.

The syndrome generator receives the BCH code from the encoder and is multiplied with H and followed by modulo two addition [3].

ISSN: 2582-3825 (online)
Submitted: 22.12.2020
Accepted: 12.01.2021
Published: 19.01.2021

Journal of Electronics and Informatics (2020)
Vol.02/ No.04
Pages: 220-227
https://www.irojournals.com/iroei/
DOI: https://doi.org/10.36548/jei.2020.4.004

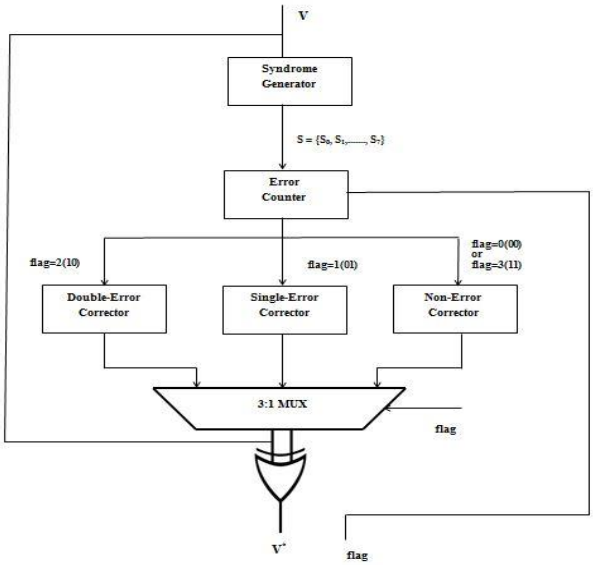

Figure 3 Block diagram of DEC-TED BCH decoder using Adaptive error correction

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

In this work the generated syndrome is

$$S = [s_7, s_6, s_5, s_4, s_3, s_2, s_1, s_0]$$

The generated syndrome vector is given to the error counter, which count the number of errors present in the received code word. The error counter generate flag signal depends on the number of errors. Table 2 Shows the flag condition for the number of errors.

Table 2
Flag condition for the number of error correction

Number of errors	2-bit flag condition flag[1]	flag[0] =
Non	0	0
Single-bit	0	1
Double-bit	1	0
Triple bit	1	1

Depending upon the flag signal produced, we can select either the single error corrector (SEC) or the double error corrector (DEC). Figure 4 [1] shows the hardware structure of error counter. The error counter is basically a AND functionality in the programming sense. The single error correction is achieved by complementing the erroneous bit [6]. This is because of error on the bit 1 is 0 and error on the bit 0 is 1. Double bit error correction is performed by comparing the syndrome vectors with each row of the parity check matrix.

ISSN: 2582-3825 (online)
Submitted: 22.12.2020
Accepted: 12.01.2021
Published: 19.01.2021

Journal of Electronics and Informatics (2020)
Vol.02/ No.04
Pages: 220-227
https://www.irojournals.com/iroei/
DOI: https://doi.org/10.36548/jei.2020.4.004

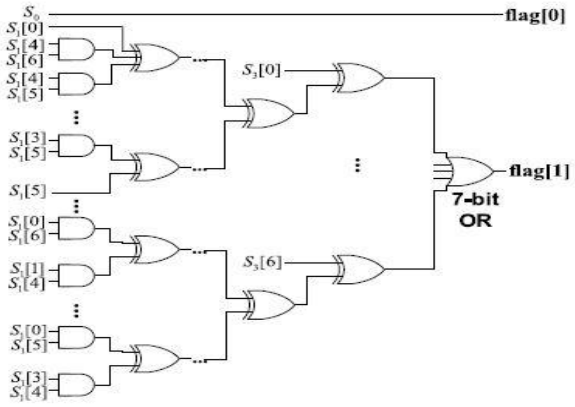

Figure 4 Error Counter

C. Invalid Transition Inhibition Technique for DEC-TED BCH code

In fully parallel BCH decoder, syndrome vector has a strong dependence in the decoder succeeding the syndrome generator block. The blocks succeeding syndrome generators are the error corrector, the parallel Chien search (LUT) and the coefficient calculator. This dependence is the reason behind the high power consumption in this decoder. When a new data is given to the decoder, before the output settle down the syndrome generator makes invalid glitches. These glitches create unwanted transitions in the block succeeding the syndrome generator, this is called invalid transition and it increase the power utilization [1].

When continuously non error code words are entered into the decoder, the impact of invalid transition in the power consumption is strong. When a non-error code word is appeared on decoder, the syndrome-coefficient of reverse error location polynomial- and error vector are settled to 0. When the following non error code word is instantly comes, the syndrome vector switch many times till it down to 0. This makes invalid transition in the blocks succeeding the syndrome generator. This invalid transition problem can be eliminated by delivering stable syndrome to the following blocks.

The obtained syndrome vectors should be fetched to the SE or DE corrector, to prevent this invalid transition. A block diagram of the proposed DEC-TED decoder using invalid transition inhibition techniques and adaptive error correction is shown in Figure 5.

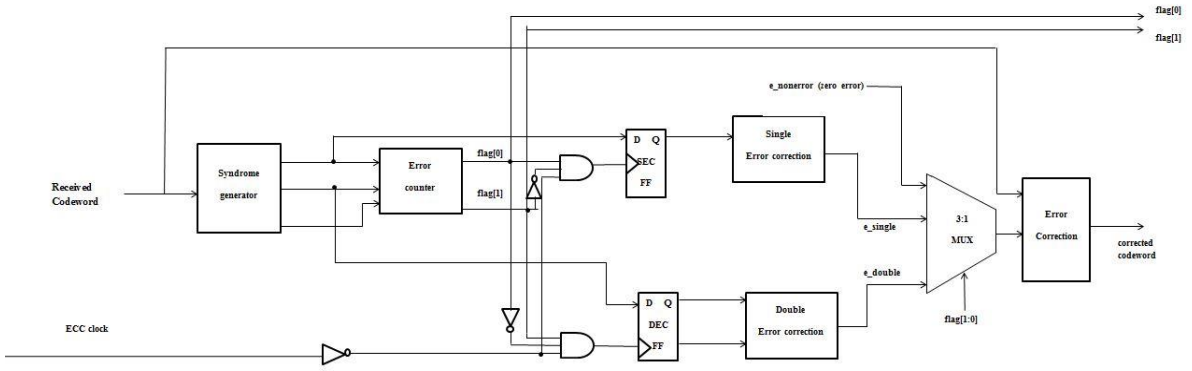

Figure 5 Block diagram of DEC-TED Decoder

After the flag bit and the syndrome vector become stable, both the Flip Flops convey the settled syndrome vectors and their control signals would be activated. For this purpose, a particular clock called the Error Correcting Clock (ECC clock) is adopted in the decoder to bring about the control signals. An inverting error correction clock is used for positive edge triggered Flip Flops.

The pulse width of the Error Correcting Clock would be greater than the sum of the delay of error counter (T_{EC}) and the worst delay of syndrome generator (T_{synd}). With the inverter and AND gates a Clock-gating technique is used to stop the simultaneous working of Single Error Corrector Flip Flops and Double Error Corrector Flip Flop. This technique is applied to the Flip Flop flag bits and clock signal. In the case of non or triple bit error, the Flip Flops do not convey any syndrome to the succeeding blocks.

ISSN: 2582-3825 (online)
Submitted: 22.12.2020
Accepted: 12.01.2021
Published: 19.01.2021

Journal of Electronics and Informatics (2020)
Vol.02/ No.04
Pages: 220-227
https://www.irojournals.com/iroei/
DOI: https://doi.org/10.36548/jei.2020.4.004

At a time only one Flip Flops transfer the settled syndrome vector to their respective corrector block (SEC or DEC), thereby reduce the power consumption. For single bit error SEC Flip Flop transfer the syndrome vector towards the SE corrector and for double bit error DEC Flip Flop transfer the syndrome vector towards the DE corrector.

IV. SYNTHESIS RESULTS AND COMPARISON

The section introduces the synthetic results of the proposed DEC-TED BCH. The DEC-TED BCH Encoder and Decoder described in Verilog HDL and simulated in xilinx FPGA using ISE design suite 14.5. The proposed invalid transition inhibition techniques and adaptive error correction are applied to the decoder. (95, 84, 7) DEC-TED Decoder (Encoder) synthesized using Xilinx FPGA Design suite 14.5. The 84 bit data is distributed over the BCH codes bch_{c1} *to* bch_{23}. The syndrome generator generates 8 syndrome vectors each of 12 bit. The error counter generates the flag signal, *flag*0 and *flag*1. After the single bit error correction and double bit error correction the input data is retrieved from the BCH encoded data through error correction block.

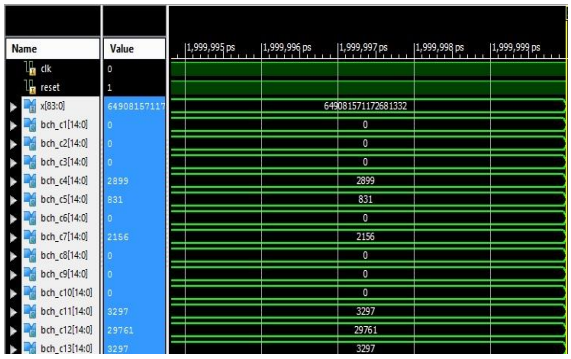

Figure 6 Simulation Result of BCH Encoder

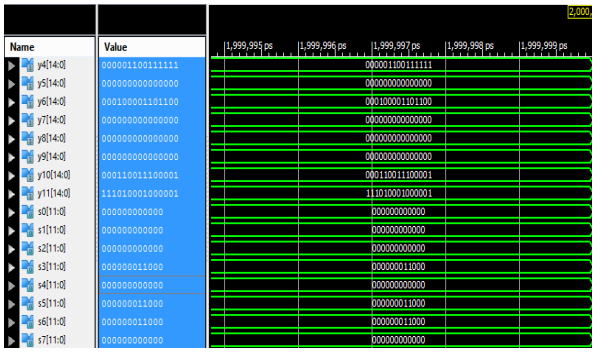

Figure 7 Simulation result of Syndrome Generator

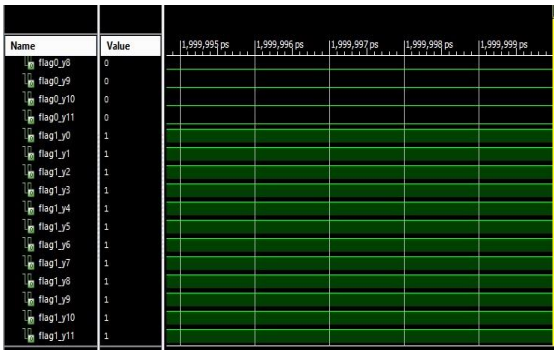

Figure 8 Simulation Result of Error Counter

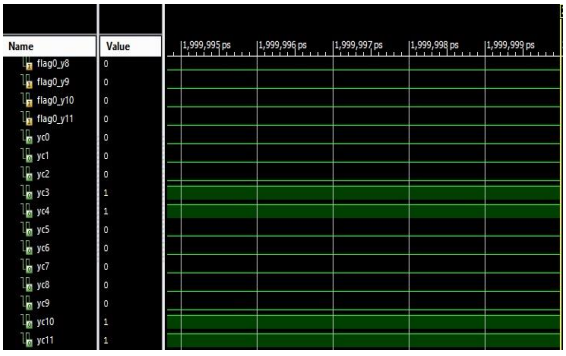

Figure 9 Single Error Corrector

Figure 10 Simulation of Double Error Corrector

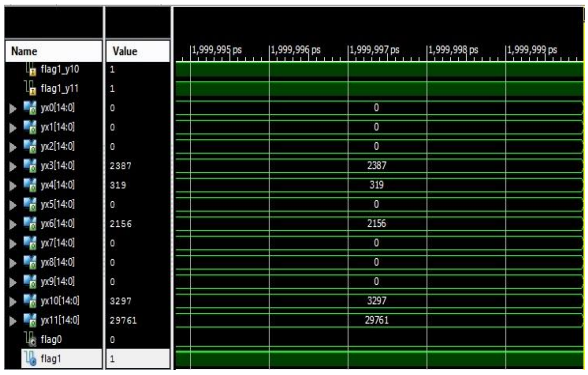

Figure 11 Simulation Result of Multiplexer

ISSN: 2582-3825 (online)
Submitted: 22.12.2020
Accepted: 12.01.2021
Published: 19.01.2021

Journal of Electronics and Informatics (2020)
Vol.02/ No.04
Pages: 220-227
https://www.irojournals.com/iroei/
DOI: https://doi.org/10.36548/jei.2020.4.004

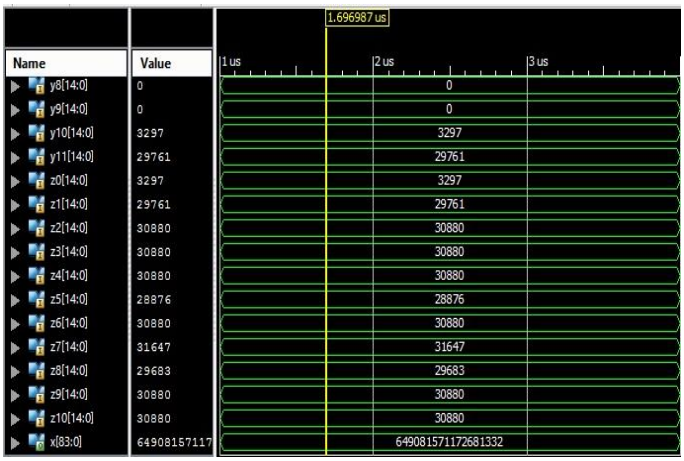

Figure 12 Simulation Result of Error Correction

In the area sense Number of slice register is 1, 1445 slice LUTs, 600 Occupied slices and 170 bonded IOB's are used. The average fan out is 4:21. The delay of this decoder is 97.312ns and the power is 81mW.

V. CONCLUSION

This brief has presented a DEC-TED BCH Encoder and (95,84,7) Emerging memories with low power in Xilinx FPGA using Design suite 14:1 and also presented an adaptive error correction technique. An invalid transition inhibition technique was developed by using flip flops to reduce power consumption in fully parallel BCH decoders. The delay for the decoder is 97.312ns and power consumption 81mW. The decoder has achieved better performance with regard to area, delay and power.

REFERENCES

[1] Sara Choi,Hong Keun Ahn, Byung kyu Song, Jung Pill Kim, Seung H. Kang, Seong-Ook Jung, *A Decoder for short BCH Codes With High Decoding Efficiency and Low Power for Emerging Memories*,IEEE Transaction on Very Large Scale Integration Systems., volume:27, Issue 2, Feb.2019

[2] K Mohana Krishna and Mrs.A.Maria Jossy Fault Tolerent *Parallel Filters Based on BCH Codes*, Int. Journal of Engineering Research Applications.,volume 5,Issue4 part 7 Apr 2015.

[3] Zhen Gao, Pedro Reviriego, Wen Pan,Zhan Xu,Ming Zhao,Jing Wang, and Juan Antonio Maestro, *Fault Tolerent Parallel Filters Based on Error Correction Codes*, IEEE Trans. Very Large Scale Integration Systems. Feb 2014.

[4] C.C. Chu,Y.M. Lin and C.H Chang,*A fully parallel BCHcodec with double error correcting capability for NOR flash application*.,IEEE Int Conf.Acoust.,Speech,Signal Process (ICASSP), Mar.2012,pp 1605-1608

[5] Livio Baldi and Gurtej Sandhu,*Emerging Memories*.,IEEE Trans., 2014,pp 30-36

[6] Riaz Naseer and Jeft Draper,*Parallel Double Error Correcting Code Design to Mitigate Multi-Bit Upsets in SRAMs*., Proc. Eur. Solid-State Circuits Conf.(ESSCIRC), Sept.2008,pp 222-225

ISSN: 2582-3825 (online)
Submitted: 22.12.2020
Accepted: 12.01.2021
Published: 19.01.2021