IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
ISSN: 2319 – 4200, ISBN No. : 2319 – 4197 Volume 1, Issue 5 (Jan. - Feb 2013), PP 17-24
www.iosrjournals.org

A High- Speed LFSR Design by the Application of Sample Period Reduction Technique for BCH Encoder

Manikandan.S.K[1], Sharmitha.E.K[2], Nisha Angeline.M[3], Palanisamy.C[4]

[1]*(Assistant Professor (Sr.Gr.) EEE, Velalar College of Engineering and Technology/ Anna University, India)*
[2]*(Student, ME VLSI Design, Velalar College of Engineering and Technology/ Anna University, India)*
[3]*(Assistant Professor (Sr.Gr.) ECE, Velalar College of Engineering and Technology/ Anna University, India)*
[4]*(Professor and Head, Department of IT, Bannari Amman Institute of Technology/ Anna University, India)*

Abstract: *Error correction is one of the important technique for detecting and correcting errors in communication channels, memories etc., BCH codes are widely been used for error detection and correction. The generated check bits of the BCH encoder along with the message bits called codeword is sent to the receiver to detect any error during the transmission. One of the main components of BCH encoder is LFSR (Linear Feedback Shift Register). LFSR find its wider application in Built-in-Self-Test, signature analyzer etc., whereas here it is used to form parity bits to concatenate with message bits for the formation of a codeword. The main advantage of LFSR is that it is simple to construct and it operates at very high clock speed, but its main drawback is that the inputs are given in bit serial. To overcome these drawbacks, DSP algorithms such as unfolding and parallel processing can be used by selecting the unfolding factor based on some design criteria. Selecting a better unfolding value reduces the sample period, decreases the clock cycle and increases the speed.*

*.**Keywords-** Bose Chaudhuri-Hocquengham (BCH), Cyclic Redundancy Check (CRC), Computational Time (CT), Galois Field (GF), LFSR, unfolding, sample period reduction.*

I. Introduction

The main usage of error correcting code is to detect and then correct the errors that are introduced by the transmission channel and storage devices. Errors are introduced from source to receiver during transmission, when the communication channel is affected by the channel noise. Error detection is a technique for detecting and correcting those errors, which are induced by the channel noise.

There are various codes available for error detection and correction. The codes are broadly classified into two types; they are a).block codes and b).conventional codes. Cyclic code is one of the classifications of block codes. The subset of the block code is BCH code.BCH code initially forms a generator polynomial by the use of finite field (GF) concept [1] and generates a parity (check) bits to be appended to the message bits to form a codeword [2]. The main component of the encoder is simply a LFSR for generating the parity bit. The components used to form LFSR are simply registers and exor gates. Series combination of both registers and exor gates forms a LFSR. The main advantage of LFSR is it is simple to construct and it operates at very high clock speed, But the main drawback of the LFSR is that the bit stream applied to LFSR should be in serial. Hence, high-speed data transmission cannot be made possible. In order to increase the throughput and speed, parallel processing can be applied by unfolding concept. Parallel processing increases the number of message bits to be processed in a clock cycle (sample rate), but increases the area also.

Unfolding is a transformation technique, which describes J consecutive iterations of the original DSP program. Unfolding increases the Iteration bound T_∞ to JT_∞. In order to reduce the sampling period it is important to calculate the iteration bound before unfolding the system to select the unfolding factor. Many important cases such as CT $> T_\infty$ and T_∞ is not an integer must be analyzed before selecting the unfolding factor. The selected unfolding value must make CT $< T_\infty$ and T_∞ is an integer, which automatically reduces the sampling period. Unfolding is a transformation technique that can be applied to any DSP program to create a new program , which describes more than one iteration of the original program [3]. Large number of iterations of an original program can be made by unfolding it by an unfolding factor.

The rest of the paper is organized as follows. Section II gives the brief summary of the Existing System. Section III contains the design procedure of LFSR for BCH (31, 16) and criteria for selecting the unfolding factor to propose a new unfolded structure to reduce the sample period and gives the steps for unfolding the DFG of the LFSR. Section IV contains the algorithm for unfolding and proposed a new architecture for LFSR. Section V analyses the data flow ,area and clock cycle of LFSR for different unfolding factors .Finally future enhancements and concluding remarks are given in section VI and VII.

II. Existing System

There are various recursive formulas developed in past to achieve the parallel architecture for CRC hardware [4]. High-speed architectures for parallel long BCH encoders are developed in [5] for a particular generator polynomial of lower order. However, the unfolding factor is not selected by analyzing various cases. Resource Sharing and power optimization techniques are applied in [4] to achieve low power high-throughput BCH error correction in VLSI for multi-level cell NAND flash memories. Novel look-ahead techniques can be used to improve the throughput for the generator polynomial of lower order [6] without considering the important criteria for selecting the unfolding factor. Retiming and unfolding of CRC architectures are introduced for lower order generator polynomial to increase the speed [7] without selecting the unfolding factor by considering some important criteria. The normal architecture and the unfolded architecture [7] are shown in Fig 1 and 2

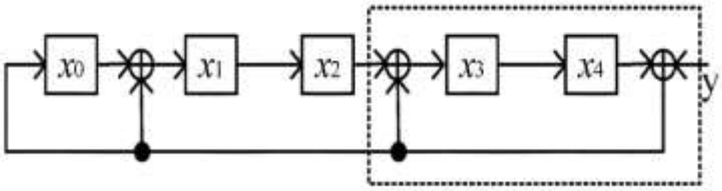

Fig 1: CRC architecture for g(x) = 1+ y+ $y^3 + y^5$

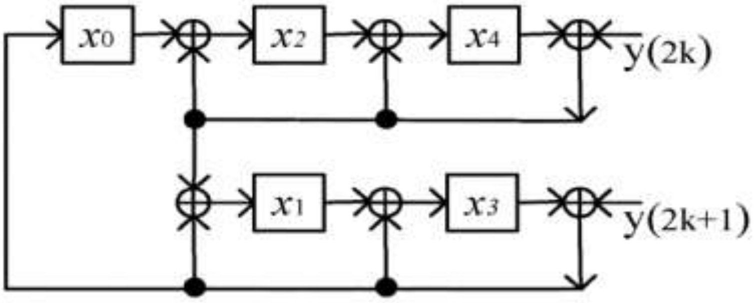

Fig 2: Two-Parallel CRC architecture for g(x) = 1+ y+ $y^3 + y^5$ for J = 2

III. Proposed System

The basic architecture of the proposed system is unfolded LFSR, for generating the parity bit. Compare to normal LFSR, the proposed unfolded LFSR architecture uses sample period reduction technique to achieve more speed. In order to achieve this objective, some important criteria has to be considered for selecting the unfolding factor to improve the design methodology. The comparison table for the proposed system and conventional system is tabulated in section V; area analysis of the proposed and conventional LFSR is tabulated in Table 4 and 5 to analyze the depth on the hardware overhead. Different levels of unfolding factors are introduced to check the hardware complexity and speed of the design.

3.1 Design of LFSR for BCH(31,16)

BCH codes are subset of the Block codes. BCH codes belong to a powerful class of multiple error correcting codes [8]. BCH codes are based on well-defined mathematical properties. These mathematical properties are based on the Galois Field or finite fields. The Finite field has the property that any arithmetic operations on field elements always have results in the field only [1]. To provide an excellent error correcting capability, the roots of the generator polynomial of the BCH codes have to be specified carefully. With a generator polynomial of g(x), a t-error correcting cyclic codes is the binary BCH codes, with a condition that g(x) must be the least degree polynomial over Galois Field GF(2). Steps for designing the generator polynomial of BCH (31,16) is explained below,

i). Choose an irreducible polynomial p(x) = $x^5 + x^2 + 1$
ii). Construct GF (2^5)
iii). Construct the minimal polynomial using the relation $\varphi_\beta(X) = \prod_{i=0}^{l-1}(x + \beta^{2^i})$ (1)

α: $x^5 + x^2 + 1$ = m1(x)
α^3: $x^5 + x^4 + x^3 + x^2 + 1$ = m2(x)
α^5: $x^5 + x^4 + x^2 + x + 1$ = m3(x)
where β be a non-zero element of GF(2^m).
iv). Form the generator polynomial using the relation g(x) = LCM(m1(x), m2(x), m3(x)). (2)

In this proposed work, BCH (31, 16) is taken as an example and an encoder is designed for it using the generator polynomial g(x)=$x^{15} + x^{11} + x^{10} + x^9 + x^8 + x^7 + x^5 + x^3 + x^2 + x$+1 and LFSR is unfolded by an unfolding factor which is selected based on some design criteria discussed in theorem[3] to improve the design methodology. LFSR architecture for BCH encoding is shown in Fig 3. Initially the 16-bit information must be made equal to the degree of the generator polynomial.

Hence, the resultant message bit is 22 bit long, which is divided by the generator polynomial to form parity bits. The parity bits that are obtained from LFSR is 100101000100010. This is systematic encoding, because information and check bits are arranged together so that they can be recognized in the resulting codeword.

General equation for codeword is,

$$i(x).x^{n-k}=q(x).g(x) + r(x) \quad (3)$$

Where,
i (x):information bit polynomial.
q (x):quotient bit polynomial.
g (x):generator polynomial.
r (x):remainder polynomial.
Encoder of BCH(31,16) comprises of parallel in serial out shift register followed by the LFSR.

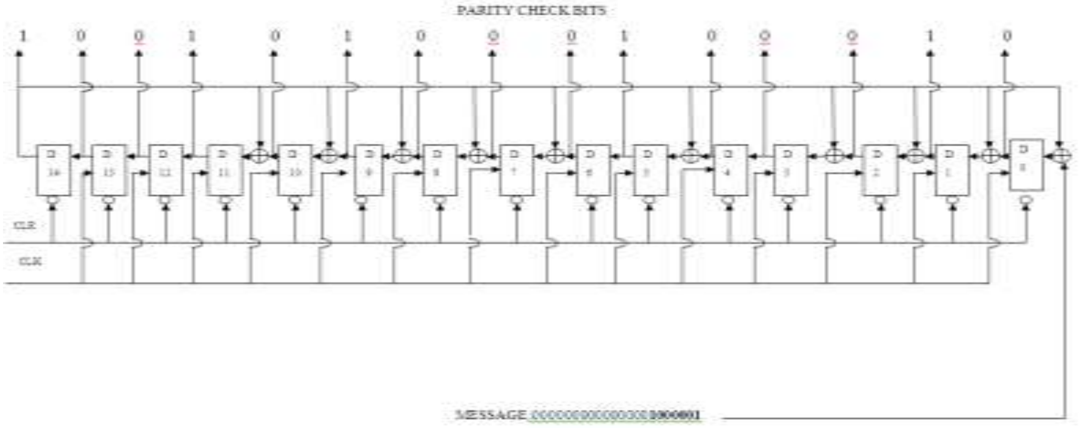

Fig 3: LFSR Architecture for g(x) = $x^{15} + x^{11} + x^{10} + x^9 + x^8 + x^7 + x^5 + x^3 + x^2 + x$+1

3.2 Unfolding LFSR

Unfolding algorithm is applied only to the LFSR architecture in this proposed design to increase the speed by reducing the sampling period.
The steps to be followed to unfold the encoder are,
i). Convert normal LFSR into DFG
ii). Calculate the iteration bound for the DFG
iii). If T_∞< CT of a node, apply unfolding to make the sampling period to be equal to T_∞. This technique is called as sample period reduction.

3.2.1 Formation of DFG for the LFSR

Often a DSP program is represented using the DFG. Here the nodes represent the computation and each of the node has its own computation time. The communication between the nodes is represented using edges. The DFG for the LFSR is shown in Fig 4.

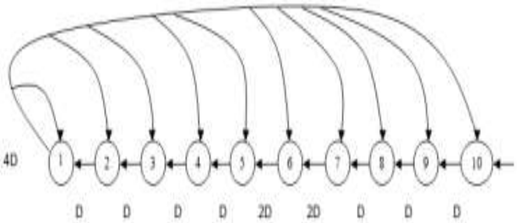

Fig 4: DFG of LFSR for g(x) = $x^{15} + x^{11} + x^{10} + x^{9}+x^{8} + x^{7} + x^{5} + x^{3} + x^{2} + x$+1

3.2.2 Calculation Of Iteration Bound

Many of the DSP algorithms contain feedback loops, which impose an inherent fundamental lower bound on the achievable iteration or sample period. This bound is referred to as iteration bound. Iteration bound is the representation of the algorithm in the form of a DFG. Same algorithm but with different representations lead to different iteration bound. Iteration bound is defined as:

$$T_\infty = \max_{l\in L}\{\frac{t1}{w1}\} \qquad (4)$$

Iteration bound is the maximum of loop bound

$$T_\infty = max\{\frac{1}{4},\frac{2}{5},\frac{3}{6},\frac{4}{7},\frac{5}{8},\frac{6}{10},\frac{7}{12},\frac{8}{13},\frac{9}{14},\frac{10}{15}\}$$

$$T_\infty = \frac{10}{15} = \frac{2}{3}\,.$$

3.2.3 Sample Period Reduction

The loop 10 ⟶ 9 ⟶ 8⟶ 7 ⟶ 6 ⟶ 5 ⟶ 4 ⟶ 3 ⟶ 2 ⟶ 1 has the maximum loop bound. Synthesis report reveals that all the nodes of the DFG has CT of 1u.t. Since T_∞< CT, iteration period cannot be made equal to T_∞. In such a case retiming can be applied but it cannot be used to reduce the CT of the critical path of the DFG to T_∞. Selection of the unfolding factor is an important criterion in sample period reduction. Unfolding factor is chosen using the relation, J=$\lceil \frac{t_u}{T_\infty} \rceil$= 2. Iteration bound of the unfolded DFG changes from $T_\infty\ to\ JT_\infty$. Where J stands for the unfolding factor. Similarly the sample period of the unfolded DFG is $\frac{T_\infty}{J}$. One more case exist is, if T_∞ is not an integer. The LFSR of g(x) $= x^{15} + x^{11} + x^{10} + x^{9}+x^{8} + x^{7} + x^{5} + x^{3} + x^{2} + x$+1 satisfies both the cases. Because its CT is greater than the iteration bound and the iteration bound is not an integer. Hence J must be selected in such a way that JT_∞ is an integer and JT_∞> node CT. The only value of J that satisfies both the condition is 3. This is clearly specified by a theorem[3].

IV. Unfolding Algorithm

It is a transformation technique that can be applied to a DSP program in order to create a new program describing more than one iteration of the original program . Unfolding a DSP program is done by selecting an unfolding factor J, which describes J consecutive iterations of the original program. Loop unrolling is also called as unfolding [8].

4.1 Algorithm Steps

1. For each node U in the original DFG, draw J nodes $U_{0,}U_1, U_{2,..........,}U_{J-1}$
2. For each edge U ⟶ V with w delays in the original DFG, draw the J edges $U_i \longrightarrow U_{(i+w)\%J}$ with $\lfloor \frac{i+w}{J} \rfloor$ delays for i = 0,1,……J-1. (5)

By this technique the speed of the LFSR is increased automatically by reducing the clock cycle. The main drawback of unfolding is that the area of the system increases and choosing a large value of unfolding factor leads to hardware complexity. After applying the unfolding technique with unfolding factor J=3 and 2 for Fig 3, three parallel and two parallel architectures are obtained and it is shown in Fig 5 and 6.

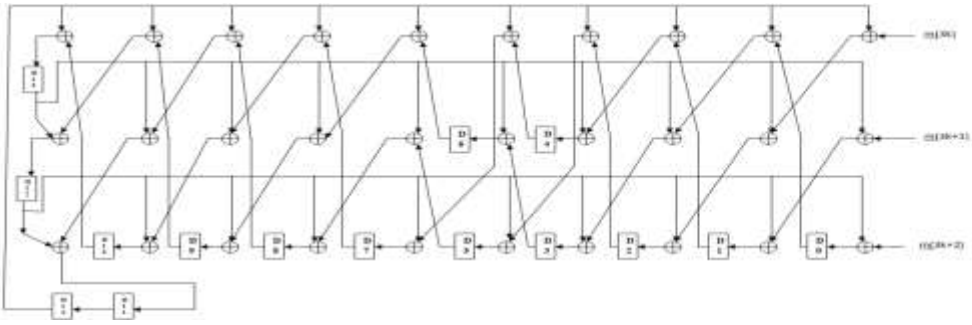

Fig 5: Three parallel LFSR for g(x) $= x^{15} + x^{11} + x^{10} + x^{9} + x^{8} + x^{7} + x^{5} + x^{3} + x^{2} + x + 1$ after Unfolding by a factor of 3

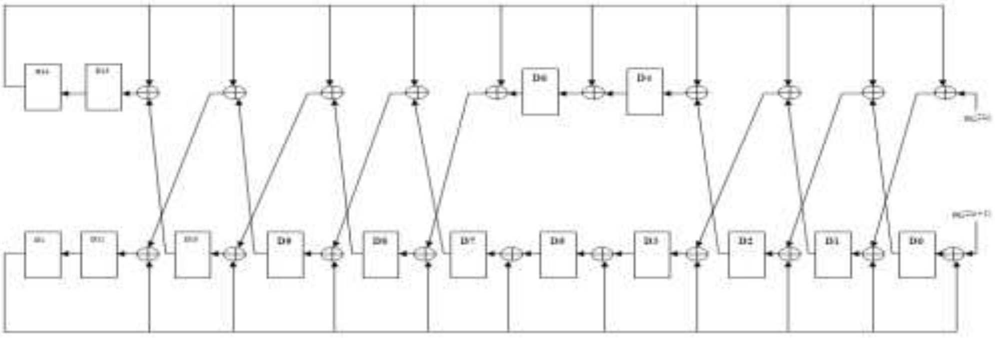

Fig 6: Three parallel LFSR for g(x) $= x^{15} + x^{11} + x^{10} + x^{9} + x^{8} + x^{7} + x^{5} + x^{3} + x^{2} + x + 1$ after Unfolding by a factor of 3

After the application of unfolding the sample period is reduced and the iteration bound is increased from 0.66 to 1.32. So that J T_∞> CT. This sample period reduction is one of the application of the unfolding algorithm.

V. Results And Discussion

Initially the codeword is formed by the generation of parity bits 100100010100010. This parity bit formation is coded in VHDL. Each of the unfolded architecture is coded in VHDL , simulated and implemented using Xilinx92i to analyze the area and speed. For the message bits:0000000001000001 and for the generator polynomial g(x)=$x^{15} + x^{11} + x^{10} + x^{9} + x^{8} + x^{7} + x^{5} + x^{3} + x^{2} + x + 1$ the normal BCH encoder simulation result is shown in Fig 7. The same architecture but with unfolding factor of 2 and 3 is simulated and verified as shown in Fig 8 and 9. From the results shown in Table 1, 2, 3 and 4, it is clear that the clock cycle decreases from 22 to 8 because of sample period reduction technique. Hence unfolding speed up the LFSR operation by decreasing the clock cycle. As far the memory is concerned, the error detection and correction must not take much time because it decreases the throughput of the system.

5.1 Data Flow Table

Table 1: Data flow table for normal LFSR

clock	Message bit	y(14 to 0)
1	1	000000000000001
2	0	000000000000010
3	0	000000000000100
4	0	000000000001000
5	0	000000000010000
6	0	000000000100000
7	1	000000001000001
8	0	000000010000010
9	0	000000100000100
10	0	000001000001000

11	0	000010000010000
12	0	000100000100000
13	0	001000001000000
14	0	010000010000000
15	0	100000100000000
16	0	000110110101111
17	0	001101101011110
18	0	011011010111100
19	0	110110101111000
20	0	101010101011111
21	0	010010100010001
22	0	100101000100010

Table 2. Data flow table for LFSR with unfolding factor 3

clock	m(3k) m(3k+1) m(3k+2)	y(14 to 0)
1	100	000000000000100
2	000	000000000100000
3	100	000000100000100
4	000	000100000100000
5	000	100000100000000
6	000	011011010111100
7	000	010010100010001
8	000	100101000100010

Table 3: Data flow table for LFSR with unfolding factor 2

Clock	m(2k) m(2k+1)	y(14 to 0)
1	10	000000000000010
2	00	000000000001000
3	00	000000000100000
4	10	000000010000010
5	00	000001000001000
6	00	000100000100000
7	00	001000001000000
8	00	000110110101111
9	00	011011010111100
10	00	110010101011111
11	00	100101000100010

5.2 Area and Clock Cycle Analysis

The design is analyzed for different levels of unfolding factors in order to discuss the hardware overhead involving in different parallelism levels. The area analysis and device utilization analysis is done by implementing this LFSR in xilinx9.2i is tabulated in Table 4 and 5

Table 4: Area and Clock Cycle Comparison table

Unfolding factor	Number of gates		Number of message bits processed per Clock cycle	Clock cycle
	combinational (Exor)	sequential (Register)		
J=0	10	15	1	22
J=2	20	15	2	11
J=3	30	15	3	8

Table 5: Device Utilization Comparison Table

Unfolding factor	Area occupied by external IOB's	Area occupied by G CLK
J=0	11%	4%
J=2	12%	4%
J=3	12%	4%

5.3 Screen Shots

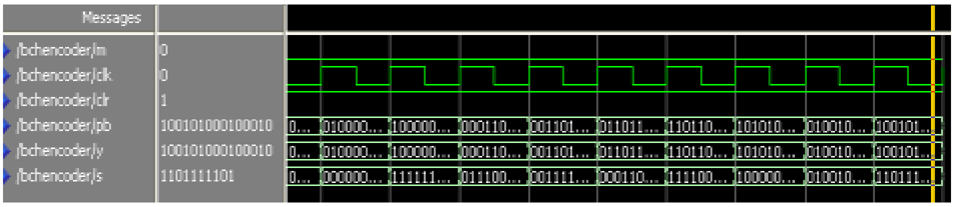

Fig 7: LFSR before Unfolding

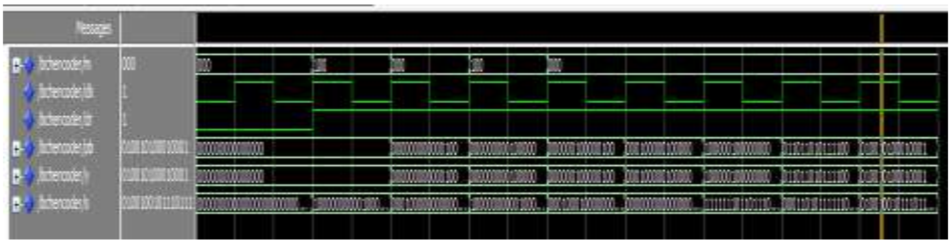

Fig 8: LFSR after Unfolding with J=3

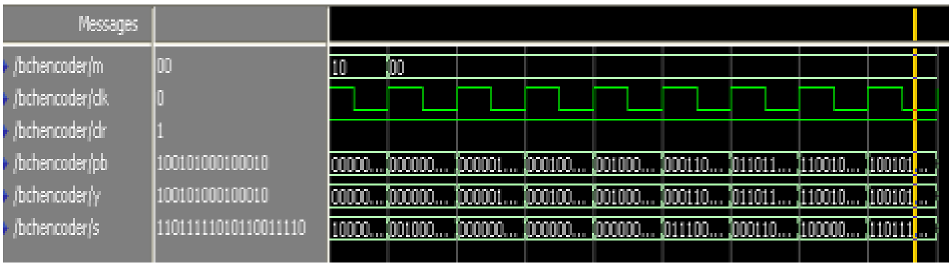
Messages
/bchencoder/m 00
/bchencoder/clk 0
/bchencoder/clr 1
/bchencoder/pb 100101000100010
/bchencoder/y 100101000100010
/bchencoder/s 1101111101011001110

Fig 9: LFSR after Unfolding with J=3

VI. Conclusion

Since the communication channels need high speed data transmission, a high throughput encoder is designed by unfolding the LFSR of the BCH encoder by checking design criteria for selecting the unfolding factor .Moreover area and clock cycle is analyzed by simulating the design in ModelSim tool by VHDL language and implemented the design in Xilinx9.2i.The obtained results reveal that unfolding increases the throughput, this in turn decreases the clock cycle which automatically increases the speed but it increases the area.

VII. Future Work

Different-pipelining techniques can be introduced to reduce the critical path of the encoder of BCH. Retiming also can be applied to further increase the speed and to reduce the power consumption and area.

Acknowledgement

The authors acknowledge the contributions of the students, faculty of Velalar College of Engineering and Technology for helping in the design of test circuitry, and for tool support. The authors also thank the anonymous reviewers for their thoughtful comments that helped to improve this paper**.** The authors would like to thank the anonymous reviewers for their constructive critique from which this paper greatly benefited.

References

[1] William Stallings, “Cryptography and Network Security-Principles and Practices, Introduction to Finite Fields”, 3rd edition, 2004.

[2] Ranjan Bose, “Information Theory, Coding and Cryptography”.

[3] K.K.Parhi “VLSI Digital Signal Processing Systems-Design And Implementation”.

[4] Wei Liu, Junrye Rho, and Wongong Sung , “Low- Power High throughput BCH error correction VLSI Design for Multi-Level cell NAND Flash Memories”.

[5] Keshab K. Parhi, ”Eliminating the Fan out Bottleneck In Parallel Long Bch Encoders” in proc IEEE, vol.51.No.3, march 2004.

[6] Naresh Reddy, B.Kiran Kumar and K.monisha Sirisha,” On the Design of High Speed Parallel CRC Circuits Using DSP Algorithms” in IJCSIT, vol.3 (5), 2012.

[7] Chao Cheng and Keshab Parhi, “High-Speed Parallel CRC Implementation Based On Unfolding, Pipelining And Retiming”, in proc, IEEE, vol.53, No.10, October 2006.

[8] John G.Proakis Masoud Salehi, ”Digital-Communications-Linear block codes, cyclic codes, BCH codes, Reed-Solomon codes,” 5th Edition, 2008.