ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

FPGA Implementation of (15, 7) BCH Encoder and Decoder for Audio Message

Lakhendra Kumar
Dept. of Electronics & Communication, DSCE Bangalore, India
lakhendrak@gmail.com

Abstract

In a communication channel, noise and interferences introduce the bit errors during the transmission of the digital message. To get the error free communication, error control codes are used. This paper discusses FPGA implementation of (15, 7) BCH Encoder and Decoder for audio message transmission and reception using Verilog Hardware Description Language. Initially audio message is converted to digital data which are framed into binary data of 7 bits. These 7 bits are encoded into 15 bit code word using (15, 7) BCH encoder. If any 2 bit error occurs in any position of 15 bit code word, it is detected and corrected. This corrected data is converted back into an audio message. The decoder being implemented here is one step majority logic decoder. Conversion of audio to digital and digital to audio, both are done using on-chip audio 97 codec available on FPGA. Simulation was carried out by using Xilinx 14.2 ISE simulator, and verified results for an arbitrarily chosen message data. Synthesis was successfully done by using the RTL compiler. Finally both encoder and decoder design is implemented on Virtex v.5 FPGA kit.

Keywords: BCH encoder, BCH decoder, FPGA, RTL compiler, Verilog

Introduction

Rapid growth in internet and mobile technology, exchange of information is common practice. Information may be text, audio, video etc. form. Transmission of information through a physical medium or wireless medium, possibility is that data gets corrupted; this leads to an error in random locations of a symbol. To have a reliable communication through a communication channel that has an acceptable Bit Error Rate (BER) and High Signal to Noise Ratio (SNR) error correcting codes are used. These codes are introduced in order to detect and correct a specified number of errors which may occur during transmission of message over a communication channel [1]-[4].

There are different types of Error correction codes that are used in present digital communication system based on the type of channel noise. Few of them are Hamming code [3], Low Density Parity Check code (LDPC) [4], Bose-Chaudhuri Hocquenghem code (BCH) [5], Reed Solomon code [6], and Turbo code [7]. These codes are different from each other in their complexity and implementation. BCH codes are widely used in the areas like mobile communication, digital communications, satellite communications, optical and magnetic storage systems, and computer networks etc.

In this work (15, 7) BCH encoder and decoder are implemented on Virtex v.5 FPGA. For designing the BCH codes, Systematic codes are used. In case of systematic codes original message d(x) is as it is in the encoded word c(x). At the transmitter side using encoder circuit binary digits are encoded by appending some extra bits with message bits also known as parity bits. The parity bits and message bits together are known as 'Code word'. At the Receiver end, code word will be received and error detection and correction process is applied. This process is known as decoding. The rest of this paper will discuss about hardware implementation of BCH encoder and decoder as well as their simulation and synthesis results for FPGA.

Related work

Cyclic decoding procedures for error corrections in BCH code was discussed in [8]. They implemented binary BCH codes for 5 bit error correction with a length of 127 bits using Peterson decoding procedure. Author has shown that burst error correcting codes are less speed and more complex in hardware compared to BCH codes Application of BCH codes in authentication of binary document images is discussed in [9]. They used (7, 4) BCH encoder for encoding of a character and

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

embedded in a binary document image, each 4 bit in a character is encoded to a 7 bit. They used BCH codes to correct one bit error in any position of 7 bit data. Implementation of (7, 4) BCH encoder to correct single error in any position of 7 bits is discussed in [10]. The circuit design and simulation was carried out using Orcad version 9.1 and implemented on FPGA (Xilinx xc4013). In [11] application of BCH code for error detection and correction in memory is presented. They compared both single error correction and double error correction BCH codes using 180nm technology. The synthesis and simulations were carried out using Synopsys Design Compiler, power consumption and area of the circuit was summarized for different (n, k) BCH code cases. The result shows that power consumption and area required was less compared to single error correction. Application of BCH codes in fault tolerant method is given in [12]. They designed a 32 bit ALU, which is secure against many faults and able to correct any 5 bit faults in any positions of 32 bits input registers of ALU. The BCH codes were used to correct multiple errors. Author has shown that compared to TMR (triple modular redundancy) & Residue code, BCH codes were the better choice in estimated area. Nonlinear multi error correcting codes in the reliable MLC NAND flash memories using BCH and RS codes is discussed in [13]. The encoder and decoder architectures for nonlinear 5 bit error corrections in flash memories can be modeled in Verilog and synthesized in RTL design compiler. Proposed work discusses, FPGA implementation of (15, 7) Binary BCH Encoder and Decoder for audio message using Verilog HDL. This work aims to correct double error in any position of 15 bit code word. Initially audio message is converted to digital signal which are then used for framing of 7 bits. These 7 bits are encoded into 15 bit code word using (15, 7) BCH encoder. If any 2 bit error in any position of 15 bit code word, is detected and corrected. This corrected data is converted back into an audio message. The decoder being implemented here is one step majority logic decoder. Conversion of audio to digital and digital to audio, both are done using on-chip audio 97 codec available on FPGA. Simulation was carried out by using Xilinx 14.2 ISE simulator, and verified results for an arbitrarily chosen message data. Synthesis was successfully done by using the RTL compiler. Finally both encoder and decoder design is implemented on Virtex v.5 FPGA.

BCH Code

BCH codes can be defined by two parameters that are code size *n* and the number of errors to be corrected *t*

Block length: $n = 2^m - 1$

Number of information bits: $k \geq n\text{-}mt$

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

The generator polynomial of the code is specified in terms of its roots over the Galois field GF (2^m). Let α be a primitive element in GF (2^m).The generator polynomial g(x) of the code is the lowest degree polynomial over GF (2). Let $m_i(x)$ be the minimum polynomials of α_i, then generator polynomial G(x) can be computed

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

In this work n=15, k=7 and t=2 is considered. Hence the generator Polynomial with α, α^2 ,α^3 ,α^4 as the roots is obtained by multiplying the following minimal polynomials:

$$m_1(x) = 1+x+x^4$$

$$m_3(x) =1+x+x^2+x^3+x^4$$

Substituting $m_1(x)$ and $m_3(x)$ in equation (1) generator polynomial is obtained.

$$G(x) = LCM \{m_1(x), m_3(x)\}$$

$$G(x) = \{(1+x+x^4) (1+x+x^2+x^3+x^4)\}$$

$$G(x) =1+x^4+x^6+x^7+x^8 \quad (2)$$

To build BCH codes over GF (2^4), we need to find out the elements of GF (2^4) generated by $p(x) =1+x+x^4$ is given in Table below.

Tables:

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

power of α		polynomial	binary representation
α	=	α	0010
α^2	=	α^2	0100
α^3	=	α^3	1000
α^4	=	$\alpha + 1$	0011
α^5	=	$\alpha^2 + \alpha$	0110
α^6	=	$\alpha^3 + \alpha^2$	1100
α^7	=	$\alpha^3 + \alpha + 1$	1011
α^8	=	$\alpha^2 + 1$	0101
α^9	=	$\alpha^3 + \alpha$	1010
α^{10}	=	$\alpha^2 + \alpha + 1$	0111
α^{11}	=	$\alpha^3 + \alpha^2 + \alpha$	1110
α^{12}	=	$\alpha^3 + \alpha^2 + \alpha + 1$	1111
α^{13}	=	$\alpha^3 + \alpha^2 + 1$	1101
α^{14}	=	$\alpha^3 + 1$	1001
α^{15}	=	1	0001

(15, 7) BCH Encoder

The (15, 7) BCH Encoder is implemented with a Linear Feedback Shift Register (LFSR). (15, 7) BCH code word are encoded as follows.

$$C(x) = x^{n-k} * M(x) + b(x) \quad (3)$$

Where, Message bits $M(x) = M_0 + M_1x +...+ M_{k-1}x^{k-1}$

Code word C(x) is $c_0 + c_1x +...+ c_{n-1}x^{n-1}$

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

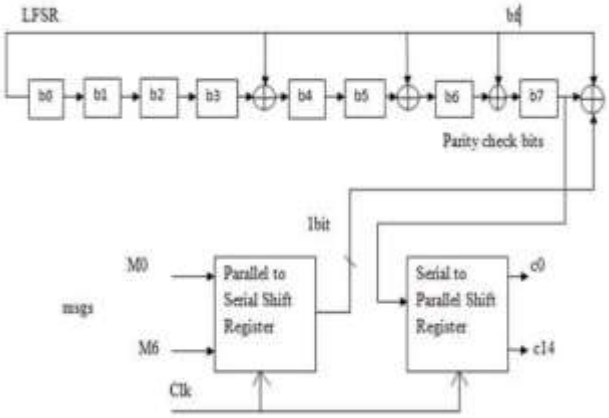
Remainder $b(x) = b_0 + b_1x + ... + b_{m-1}x^{m-1}$

also ci, Mi, bi are the subsets of Galois field.

Figure 1 shows block diagram of (15, 7) BCH Encoder module. The 7 message bits (M_0, M_1….M_6) are applied to the parallel to serial shift register. The output of parallel to serial shift register will be sent to (15, 7) BCH Encoder module as shown in figure. Using these message bits parity bits are computed and sent to serial to parallel shift register. These parity bits are appended to original message bits to obtain 15 bit encoded data. This entire encoding process requires 15 clock cycles.

Figure-1: Block diagram of (15,7) BCH Encoder

(15, 7) BCH Decoder



Given an (n, k) cyclic code C for which J parity-check sums orthogonal on an error digit can be formed, the one-step majority-logical decoding of the code can easily be implemented. First, from the null space C_d of the code, we determine a set of J vectors $w_1, w_2 \ldots w_J$ that are orthogonal on the highest-order digit position, X^{n-1}. Then, J parity-check sums A_1, $A_2 \ldots$, A_J orthogonal on the error digit e_{n-1} are formed from these J orthogonal vectors and the received vector r. The vector w_j tells what received digits should be summed up to form the check-sum A_j. The J check-sums can be formed by using J multi-input modulo-2 adders. Once these J check-sums are formed, they are used as input to a J multi-input majority-logic gate. The output of a majority-logic gate is 1 if and only if more than half its inputs are 1; otherwise, the output is 0. The output is the estimated value of e_{n-1}. A general one-step majority-logic is shown in the fig.2. This decoder is called the type-II one-step majority-logic decoder. The error correction procedure is as follows:

Step 1. With gate 1 turned on and gate 2 turned off, the received vector r is read into the buffer register.

Step 2. The J parity-check sums orthogonal on e_{n-1} are formed by summing the appropriate received digits.

Step 3. The J orthogonal check sums are fed into a majority-logic gate. The first received digit r_{n-1} is read out of the buffer and is corrected by the output of the majority logic gate.

Step 4. At the end of Step 3, the buffer register has been shifted one place to the right with gate 2 on. Now, the second received digit is in the rightmost stage of the buffer register and is corrected in exactly the same manner as was the first received digit. The decoder repeats step 2 and 3.

Step 5. The received vector is decoded digit by digit in the same manner until a total of n shifts.

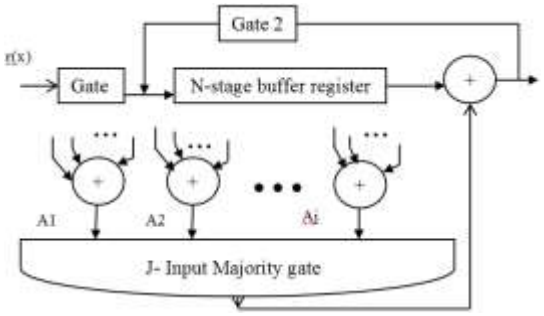


Figure-2: General one-step majority logic decoder [15]

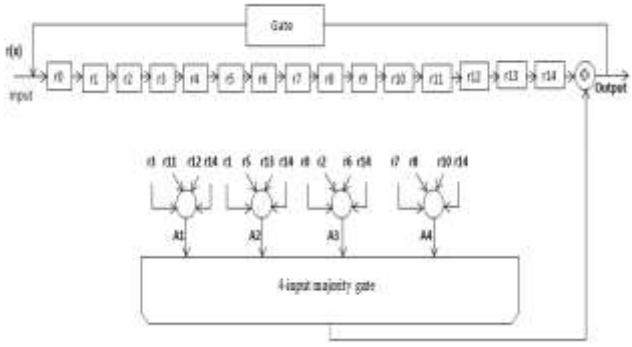


Figure-3: Type – II single-step majority-logic decoder for the (15,7) BCH code [15]

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

Hardware implementation

In this section, we have discussed the design of encoder and decoder implemented on Virtex v.5. Virtex-5 devices are offered exclusively in high performance flip-chip BGA packages that are optimally designed for improved signal integrity and jitter. Package inductance is minimized as a result of optimal placement and even distribution as well as an increased number of Power and GND pins. The **Xilinx XUPV5-LX110T** is a versatile general purpose development board powered by the **Virtex®-5 FPGA**. It is a feature-rich general purpose evaluation and development platform, includes on-board memory and industry standard connectivity interfaces, and delivers a versatile development platform for embedded applications.

BCH Encoder

Figure.2 shows block diagram of BCH encoder. Basically, encoder module is consists of three modules.

[1] 7 bit Parallel to Serial Shift Register
[2] Encoder module - Linear feedback shift register
[3] Serial to Parallel Shift Register

Steps:

Step 1. Initialize the input 'i', control signals 'start' and integer 'k', and clock 'clk'.

Step 2. If start = 0

a. Increment the value of counter 'k'.
b. If k<8,
   i. cout= i[k-1]
   ii. tout= d[0]^cout
   iii. d={tout,d[7],d[6],d[5],d[4]^tout, d[3], d[2]^tout, d[1]^tout}
   iv. c={c[13:0],cout}
c. If k ≥ 8, c={d[7:0], i[6:0]}

Step 3. If start ≠ 0

a. k = 0
b. c = 0
c. d = 0

Single Step Majority Logic Decoder

A general one-step majority-logic is shown in the figure 3. This decoder is called the type-II single-step majority-logic decoder.

Steps:

Step 1. Initialize the input 'r', control signal 'start' and clock signal 'clk'.

Step 2. If start = 0

a. Increment the value of k
   i. If k < 16
      ✓ If a == 0111 or a == 1011 or a == 1101 or a == 1111, put aout = 1 and nerr = nerr + 1
      ✓ Else reset the value of aout.
   ii. Compute cout = rout ^ aout
   iii. Assign a temporary variable t = {cout, t [14:1]}
b. Go to Step 2.a till k ≥ 16

Step 3. Display the output out = t

Setup for audio message to BCH Encoder and Decoder

This section communicates with the AC'97 Audio Codec on the Virtex v.5 development board. It implements codec configuration, audio input, and audio output capabilities. The AC'97 codec communicates with the FPGA using a serial protocol. The FPGA sends it a series of commands, which control volume, muting, record source selection, etc. It also sends and receives PCM encoded audio in this format.

In the designed communication system as shown in figure 4, we are passing audio message. ADC embedded in AC-97codec chip converts the audio into digital signal which give the output of 20 bit frame. Now framing is done in such a way that we get a frame of 7 bits which are encoded by LFSR-encoder. The encoded output is passed through noisy channel where it adds the noise. The received signal is passed through single-step majority logic decoder. Output of decoder, which is corrected one, is framed again with each frame of 20 bits. Now the frames are passed through DAC so that the DAC can gives audio output to the headphone. Here, we are receiving both transmitted voice and corrected voice in the left and right ear respectively.

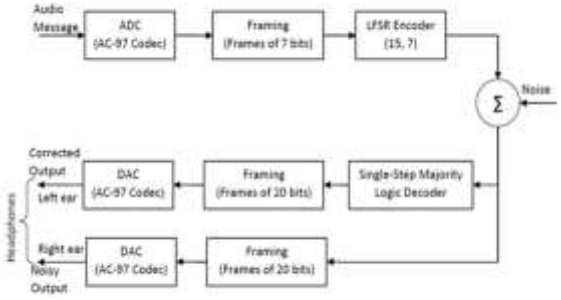


Fig.4 Design of Communication System

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

Results and discussion

In this section simulation and synthesis results of (15, 7) BCH Encoder and Decoder is discussed. The system has been simulated using Xilinx14.2 ISE simulator and functionality of encoder and decoder is verified. Synthesis was carried out by using the ISE RTL compiler and device utilization is summarized.

Simulation Results of (15, 7) BCH Encoder and Decoder with digital data

The Figure-5 shows simulation waveform of (15, 7) BCH Encoder. Here audio signal is considered as a message. As seen in figure when the start pin is high, the input is loaded to Encoder and all other intermediate signals are set to zero. When the start pin set to low, audio message being passed through codec converts into 7 bit binary digits upon framing. Using these binary digits the parity bits were calculated, meanwhile it takes 7 clocks and these parity bits are appended to the original message bits to obtain a 15 bit encoded data or code word. At the end of 8th clock, encoded data is indicated at the output terminal. The same process repeats for other digital data as well.

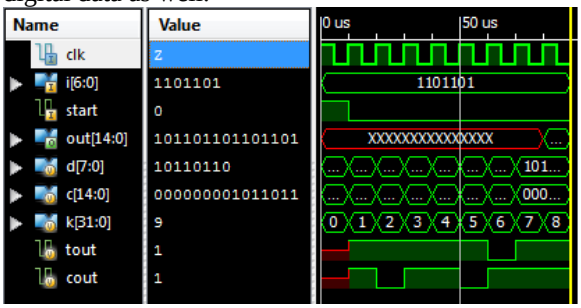


Figure-5: Simulation results for (15, 7) BCH Encoder

Input message, m(x) = '1101101'
Code word, c(x) = '101101101101101'

Parity Bits

Figure-6 shows simulation waveform of (15, 7) BCH Decoder with two bit errors. As seen in figure when the load pin is high at the first clock, encoded data or received vector r(x) is loaded as input to decoder and all other intermediate signals are set to zero. The input is given at 'r' = '101101**0**0110**0**101'. At 16th clock, the output is corrected and is given as 'out'= '101101101101101' and number of errors:nerr = 2. At the 16th clock, it indicates decoded data at the output terminal. This corrected data is converted into an audio message using Codec. The same procedure is repeated for received data as well.

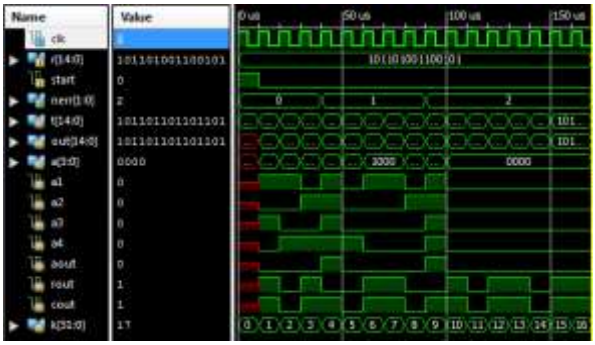


Figure-6: Simulation results for (15, 7) BCH Decoder

Synthesis Results of Communication System

Figure 7 shows the synthesis and RTL schematic of communication system with BCH encoder and single step majority logic decoder as shown in figure 4.

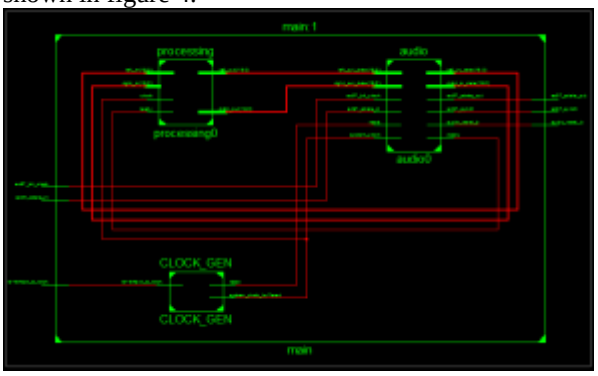


Figure-7: Audio Signal Transmission and Reception - RTL Schematic

Device Utilization Summary

Figure-8 shows the design summary for audio signal transmission and reception over FPGA.

Timing Summary for communication system

Speed Grade : -2
Minimum period : 4.889ns
(Maximum Frequency: 204.537MHz)
Minimum input arrival time before clock : 1.380ns
Maximum output required time after clock : 2.830ns

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	152	69120	0%
Number of Slice LUTs	120	69120	0%
Number of fully used LUT-FF pairs	75	197	38%
Number of bonded IOBs	6	640	0%
Number of BUFG/BUFGCTRLs	3	32	9%
Number of DCM_ADVs	1	12	8%

Figure.8 Audio Signal Transmission and Reception Design Summary

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

Conclusion

The usage of error correcting codes is very important in a modern communication system. In this paper implementation of (15, 7) BCH Encoder and Decoder for audio message is discussed. Initially audio message is converted into binary data of 7 bits using AC-97 CODEC. This 7 bit is encoded into15 bit code word. If any 2 bit error in any position of 15 bit code word, it can be detected and corrected. This corrected data is converted into an audio message. The decoder is implemented using one-step majority-logic. Simulation was carried out by using Xilinx 14.2 ISE simulator and verified results for an arbitrarily chosen message data. Also design of both encoder and decoder successfully implemented on Virtex v.5 FPGA hardware. Synthesis was successfully done by using the RTL compiler, device utilization is summarized for encoder and decoder both for Virtex v.5 FPGA.

Acknowledgements

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people who made it possible, whose proper guidance and encouragement has served as beacon light and crowned our efforts with success. I take this as an opportunity to thank all the distinguished personalities for their enormous and precious guidance throughout the duration of this paper.

I thank my guide, **Dr. K. L. SUDHA, Professor, Department of Electronics and Communication Engineering**, **Dayananda Sagar College of Engineering** for her valuable guidance and all the encouragement leading me throughout the completion of this paper.

I am indebted to all those who have directly or indirectly helped me in bringing out this paper satisfactorily in the specified duration.

References

[1] *R. C. Bose and D.K. Ray-Chaudhuri, "On a class of error correcting binary group codes", Information and control, 3: 68-79, March 1960.*

[2] *R. W. Hamming "Error Detection and Error Correction Codes" Bell Systems Tech. Journal, vol. 29, pp 147-160, April, 1950.*

[3] *Eltayeb S. Abuelyaman, Abdul-Aziz S. Al-Sehibani "Optimization of the Hamming Code for Error Prone Media", IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.3, March 2008.*

[4] *Thomas J. Richardson, M. Amin Shokrollahi, "Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes", IEEE transactions on information theory,Vol.47,No.2, February 2001.*

[5] *Amit Kumar Panda, "FPGA Implementation of Encoder for (15, k) Binary BCH Code Using VHDL and Performance Comparison for Multiple Error Correction Control", 2012 International Conference on Communication Systems and Network Technologies.*

[6] *Warren J. Gross, Frank R. Kschischang, "Applications of Algebraic Soft-Decision Decoding of Reed–Solomon Codes", IEEE transactions on communications, Vol. 54, No. 7, July 2006.*

[7] *Claude Berrou, Alain Glavieeux, "Near optimum error correcting coding and decoding Turbo codes", IEEE transactions on communications Vol.44, No.10, October 1996.*

[8] *R.T. Chien, "Cyclic decoding procedure for Bose Chaudhuri Hocquenghem Codes", IEEE Trans. On Information Theory, vol. IT-IO, pp. 357-363, October 1964.*

[9] *Da-Chun Wu, AND Ming-Kao Hsu, "Authentication of Binary Document Images Based on Embedding the BCH Codes of Watermarks", Asian Journal of Health and Information Sciences, Vol. 1, No. 4, pp. 446-455, 2007.*

[10] *Laurencin Mihai Ionesco, Constantin Anton, "Hardware Implementation of BCH Error-Correcting Codes on a FPGA", International Journal of Intelligent Computing Research (IJICR), Volume 1, Issue 3, June 2010.*

[11] *P. Reviriego, C. Aggrades, J. A. Maestro, "Efficient error detection in Double Error Correction BCH codes for memory applications", Microelectronics Reliability (2012) 1528–1530.*

[12] *Mahadevaswamy V P, Sunitha S L, B N Shobha, "Implementation of Fault Tolerant Method Using BCH Code on FPGA", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-4, September 2012.*

[13] *Zhen Wang, Mark Karpovsky, "Nonlinear Multi-Error Correction Codes for Reliable MLC NAND Flash Memories", IEEE transactions on very large scale integration*

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

(VLSI) systems, vo.20, no.7, pp.1221-1234, July 2012.

[14] S.K Moschoyiannis," Group theory and error correcting/detecting codes" master of science in information system, September 2001.

[15] S. Lin, and D.J. Costello, Jr., "Error Control Coding", Prentice-Hall, New Jersey, 1983.

Author Biblography

Lakhendra Kumar
He is a Project Scholar in the Electronics &Communication Department,Dayananda Sagar College of Engineering, Visvesvaraiya Technological University, Belgaum. He is currently pursuing Bachelor of Engineering degree from VTU. His research interests are error control coding, digital electronics and VLSI.

E-mail-id:
lakhendrak@gmail.com