FPGA IMPLEMENTATION OF (15,7) BCH ENCODER AND DECODER FOR TEXT MESSAGE

Rohith S[1], Pavithra S[2]
[1]*Assistant Professor,* [2]*PG Scholar, ECE, Nagarjuna college of engineering & Technology, Karnataka, India,*
rohithvjp2006@gmail.com**, **paviec01@gmail.com

Abstract

In a communication channel, noise and interferences are the two main sources of errors occur during the transmission of the message. Thus, to get the error free communication error control codes are used. This paper discusses, FPGA implementation of (15, 7) BCH Encoder and Decoder for text message using Verilog Hardware Description Language. Initially each character in a text message is converted into binary data of 7 bits. These 7 bits are encoded into 15 bit codeword using (15, 7) BCH encoder. If any 2 bit error in any position of 15 bit codeword, is detected and corrected. This corrected data is converted back into an ASCII character. The decoder is implemented using the Peterson algorithm and Chine's search algorithm. Simulation was carried out by using Xilinx 12.1 ISE simulator, and verified results for an arbitrarily chosen message data. Synthesis was successfully done by using the RTL compiler, power and area is estimated for 180nm Technology. Finally both encoder and decoder design is implemented on Spartan 3E FPGA.

Index Terms: *BCH Encoder, BCH Decoder, FPGA, Verilog, Cadence RTL compiler.*

----------***----------

1. 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 that data get corrupted this leads to an error in a random only selected locations of a symbol or the entire 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 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 is implemented on Sparten3E FPGA. For designing the BCH codes, two coding techniques are used. They are Systematic codes and Non systematic codes. In case of systematic codes original message d(x) is as it is in the encoded word c(x). Where as in case of non-systematic code encoded word c(x) is obtained by multiplying message d(x) with generator polynomial G(x). Hence message data will not be same in the encoded code word. 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 called as 'Codeword. At the Receiver end codeword will be received and error detection and correction process is applied. This process is known as decoding. If error presents in received data within a correction limits, the error will be corrected and original message is retrieved. This improves the quality of transmitted message to great extent and hence reduction in error rate.

The rest of this paper is organized as follows. Related work is discussed in section 2. (15, 7) BCH Encoder and Decoder are discussed section 3. In section 4 Simulation and synthesis results are discussed. Finally conclusion remarks are given in section 5.

2. 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 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 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 shown that compared to TMR (triple modular redundancy) & Residue code, BCH codes were the better choice in estimated area. Non linear 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 non linear 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 text message using Verilog HDL. This work aim to correct double error in any position of 15 bit codeword. Initially each character in a text message is converted into binary data of 7 bits. This 7 bit is encoded into 15 bit codeword. Using (15, 7) BCH Encoder if any 2 bit error in a 15 bit codeword, it will detected and corrected. The corrected data is converted into an ASCII character. The decoder is implemented using the Peterson and Zierler algorithm and chine's search algorithm. Simulation was carried out by using Xilinx 12.1 ISE simulator, and verified the results for an arbitrarily chosen message data. Synthesis was successfully carried out using the Cadence RTL compiler, power and area is estimated for 180nm Technology.

3. 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: $k \geq n\text{-}m*t$
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

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

In this work n=15, k=7 and t=2 is considered. Hence the generator Polynomial with,α, α^2,... α^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 m1(x) and m3(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+x4 is given in Table below.

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

Powers of primitive element	Binary representation	Polynomial form
α^0	0001	1
α^1	0010	α
α^2	0100	α^2
α^3	1000	α^3
α^4	0011	$\alpha+1$
α^5	0110	$\alpha^2+\alpha$
α^6	1100	$\alpha^2+\alpha^3$
α^7	1011	$1+\alpha+\alpha^3$
α^8	0101	α^2-1
α^9	1010	$\alpha+\alpha^3$
α^{10}	0111	$1+\alpha+\alpha^2$
α^{11}	1110	$\alpha^3+\alpha+\alpha^2$
α^{12}	1111	$\alpha+1+\alpha^2+\alpha3$
α^{13}	1101	$1+\alpha^3+\alpha^2$
α^{14}	1001	$1+\alpha^3$
α^{15}	1	1

3.1 (15, 7) BCH ENCODER

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

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

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

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

Remainder $b(x) = b_0 + b_1x + ... + b_{m-1}x^{m-1}$ also, c_i, j_i, b_i are the subsets of Galois field.

Figure 1 shows block diagram of (15, 7) BCH Encoder module. The 7 message bits (M0, M1….M6) 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.

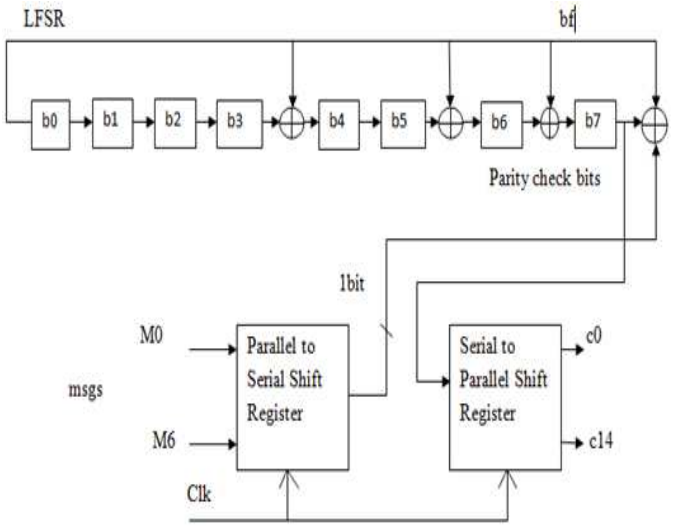

Figure-1: Block diagram of (15,7) BCH Encoder

3.2 (15, 7) BCH Decoder

The Figure-2 shows the block diagram of (15, 7) BCH decoder. The decoding algorithm for BCH codes consists of three major steps.

- Calculate the syndrome value S_i , i=1,2,….,2t from the received word r(x).
- Determine the error location polynomial $\sigma(x)$
- Find the roots of $\sigma(x)$and then correct the errors

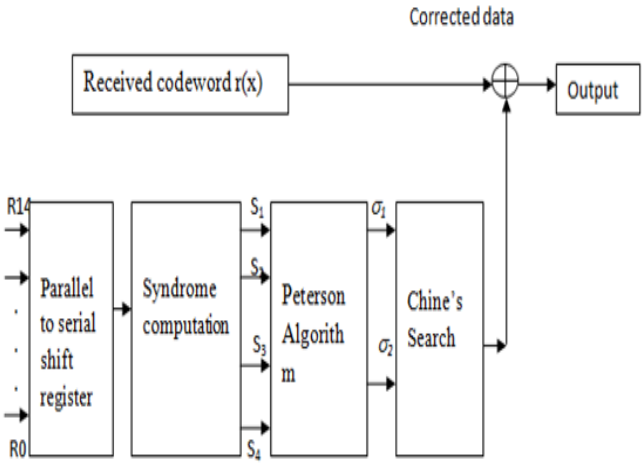

Figure-2: Block diagram for (15, 7) BCH Decoder.

The 15 bit received data is given as an input to the parallel to serial shift register, the obtained serial output will be used as a input to compute syndrome s(x) using the circuit as shown in Figure-3. If s(x) = 0, the transmission is error free. Otherwise, transmitted message will be in error. This entire process is known as error detection process.

The error correction process includes Peterson and Zierler algorithm, Chine's search algorithm. Peterson's algorithm accepts syndrome s(x) as a input and computes error locator polynomial σ(x). For finding the error locator polynomial using the formula $\sigma_1 = S_1, \sigma_2 = (S_3+S_1^3)* (S_1^{-1})$. This polynomial can be further used to find the location of the errors. Using chine's search algorithm error location is determined. This process includes searching the unique roots of the error locator polynomial. The input for the chine's search is σ(x) and it returns roots of error locator polynomial which corresponds to the error positions.

Chine's search circuit for the double error correction (15, 7) BCH code is shown in Figure-4. Using the error location information, errors will be corrected by simply flipping (converting 1 to 0 and 0 to 1) location in received 15 bit code word. The corrected data will be converted in to ASCII character. This decoding procedure will be repeated for other received code words.

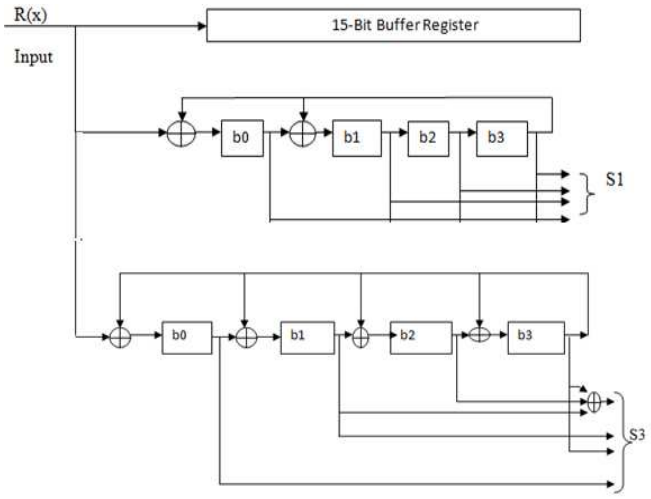

Figure-3: Syndrome circuit for (15, 7) BCH Decoder

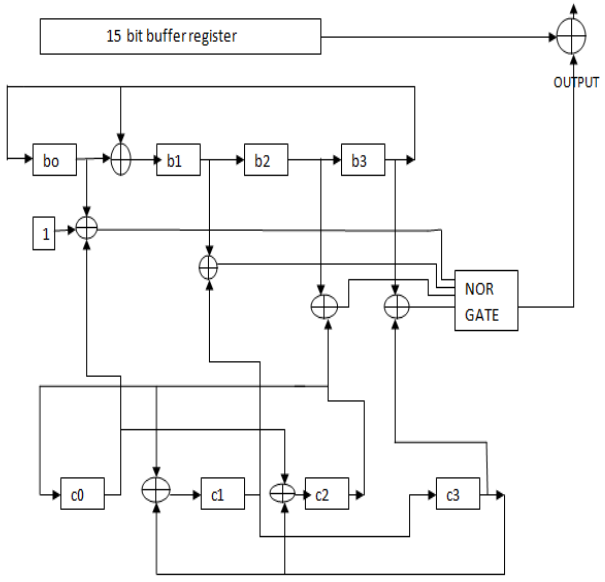

Figure-5: Chine's search circuit for the double error correction (15, 7) BCH code

4. 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 Xilinx12.1 ISE simulator and functionality of encoder and decoder is verified. Synthesis was carried out by using the cadence RTL compiler and power and area is estimated for 180nm technology.

4.1 Simulation Results of (15, 7) BCH Encoder and Decoder

The Figure-6 shows simulation waveform of (15, 7) BCH Encoder. Here "BENGALURU" is considered as a text message. As seen in figure when the reset pin is high, the input is loaded to Encoder and all other intermediate signals are set to zero. When enable pin set to high and the reset pin to low, ASCII character converted into 7 bit binary digits. Using these binary digits the parity bits were calculated and these parity bits are appended to the original message bits to obtain a 15 bit encoded data or codeword. To indicate the Encoded data in the output terminal hold pin is used. If hold pin is high 15 bit encoded data is obtained at the output port. The same process repeats for other characters as well.

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

Figure-7 shows simulation waveform of (15, 7) BCH Decoder with two bit errors. As seen in figure when the reset and load pin is high, encoded data or received vector r(x) is loaded as input to decoder and all other intermediate signals are set to zero. In the decoding process initially, enable pin set to high and the reset pin set to low. During this received 15 bit encoded data is given as input for syndrome computation. If no error in received code word syndrome output is zero. Else syndrome output will not be zero. It indicates error present in the 15 bit received data. Once error in two bits of 15 bit codeword, it is detected and corrected using Peterson and Zierler algorithm and Chine's search algorithm as discussed in section 3. To indicate status of error in Message flag bit is used. When the Flag pin is high, it indicates decoded data at the output terminal and it remains low for next corrected data available at the output terminal. This corrected data is converted into an ASCII character. The same procedure is repeated for other characters as well.

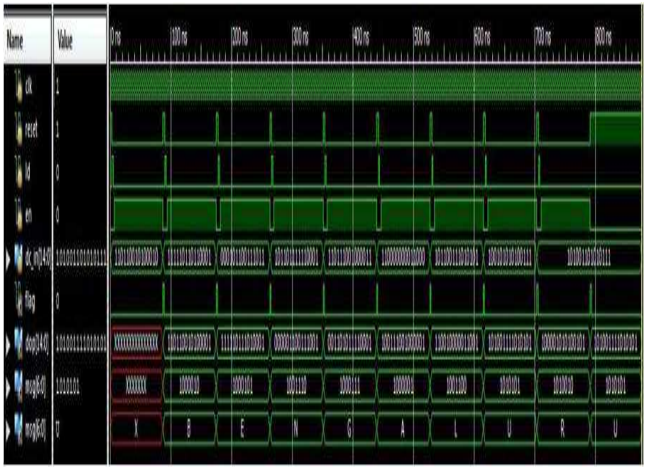

Figure-7: Simulation results for (15, 7) BCH Decoder

4.2 Synthesis Results of (15, 7) BCH Encoder and Decoder

The Figure-8 and Figure-9 shows RTL schematic of (15, 7) BCH Encoder and Decoder generated by Cadence RTL compiler. Table -2 shows power and area estimations of (15, 7) BCH Encoder for 180nm technology. **S**ynthesis was successfully carried out by using cadence RTL compiler. Power and area is estimated for 180nm technology. The power and area obtained for (15, 7) BCH Encoder are 151867.464 nw and 6961µm^2 and for Decoder are 477932.501 nw and 14051 µm^2 it is given in Table-2.

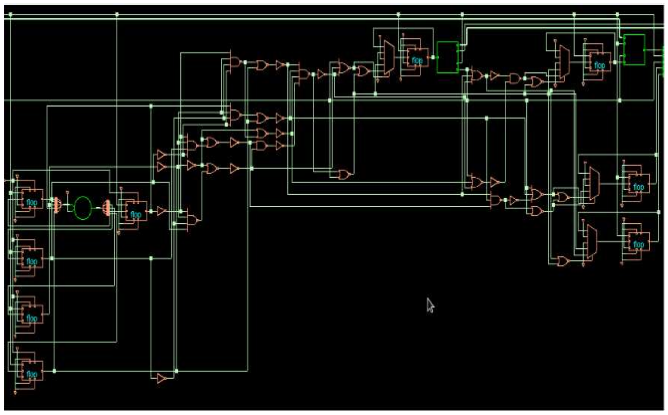

Figure-8: RTL schematic of (15, 7) BCH Encoder.

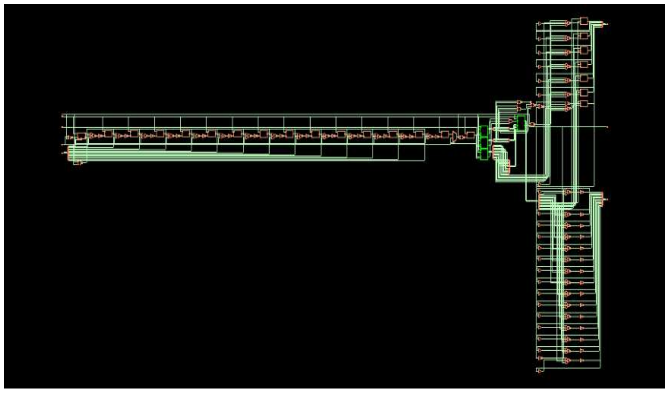

Figure -9: RTL schematic of (15,7) BCH Decoder

Table-2: Power and Area computations of (15,7) BCH Encoder and Decoder for 180nm Technology

Name of the Module	Power(nw)	Area (µm^2)
Encoder	151867.464	6961
Decoder	477932.501	14051

5. CONCLUSIONS

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 text message is discussed. Initially each character in a text message is converted into binary data of 7 bits. This 7 bit is encoded into 15 bit codeword. If any 2 bit error in any position of 15 bit codeword, it can be detected and corrected. This corrected data is converted into an ASCII character. The decoder is implemented using the Peterson and Zierler algorithm and chine's search algorithm. Simulation was carried out by using Xilinx 12.1 ISE simulator and verified results for an arbitrarily chosen message data. Also design of both encoder and decoder successfully implemented on Spartan 3E FPGA hardware. Synthesis was successfully done by using the cadence RTL compiler, power and area is estimated for 180nm technology. The power and area obtained for (15, 7) BCH Encoder are 151867.464 nw and 6961µm2 and for Decoder are 477932.501 nw and 14051 µm2.

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.
[3]. 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.
[4]. 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.
[5]. Claude Berrou, Alain Glavieeux, "Near optimum error correcting coding and decoding Turbo codes", IEEE transactions on communications Vol.44, No.10, October 1996.
[6]. R.T. Chien, "Cyclic decoding procedure for Bose Chaudhuri Hocquenghem Codes", IEEE Trans. on Information Theory, vol. IT-IO, pp. 357-363, October 1964.
[7]. 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.
[8]. 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.
[9]. P. Reviriego, C. Aggrades, J. A. Maestro, "Efficient error detection in Double Error Correction BCH codes for memory applications", Microelectronics Reliability (2012) 1528–1530.
[10]. 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.
[11]. Zhen Wang, Mark Karpovsky, "Nonlinear Multi-Error Correction Codes for Reliable MLC NAND Flash Memories", IEEE transactions on very large scale integration (VLSI) systems, vo.20, no.7, pp.1221-1234, July 2012.
[12]. S.K Moschoyiannis," Group theory and error correcting/detecting codes" master of science in information system, September 2001.
[13]. S. Lin, and D.J. Costello, Jr., "Error Control Coding", Prentice-Hall, New Jersey, 1983.

BIOGRAPHIES

ROHITH S received B.E. Degree in Electronics and Communication Engineering in 2006 and M.Tech degree in VLSI Design and Embedded systems in 2008 from Visvesvaraya Technological University, Karnataka. He is also pursuing Ph.D Degree in Visvesvaraya Technological University, Karnataka. Currently he is working as a Asst. Professor in the Department of E&C, Nagarjuna College of Engineering and Technology, Bangaluru, India. His main area of interest includes Digital Watermarking, Steganography, Error Control Coding, Cryptography and VLSI Design.

PAVITHRA S was born in Mandya. I received the B.E degree in ECE from VTU, Belgaum, Karnataka, in 2010, and M.Tech in VLSI Design and Embedded Systems from VTU, Belgaum. Her area of research includes VLSI Design, Error Control Coding.