Design and Implementation of Hamming Code on FPGA using Verilog

Ravi Hosamani, Ashwini S. Karne

Abstract — In mathematics, digital communication and information theory, error detection and correction has great practical importance in maintaining information integrity across noisy channels. Error coding is considered as a method of detecting and correcting these errors to ensure that the information is transferred intact from its source to its destination. There are various error correcting techniques to detect and correct the error. One of the popular technique based on forward error correction is Hamming Code. This paper focuses on design and its hardware implementation on Field programmable Gate Array(FPGA). The design includes both of the encoder and decoder systems to be used for the serial data transmission and reception of the wireless transceiver systems. The design has been simulated and verified using ISim simulator and Verilog HDL. Spartan-3 FPGA trainer kit for Xilinx 14.3 has been used for the implementation.

Index Terms— Error coding, Hamming code, encoder, decoder, Verilog HDL, FPGA, Xilinx, Spartan 3.

I. INTRODUCTION

In digital communication systems, environmental interference and physical defects can cause random bit errors during data transmission. Error coding is a method of detecting and correcting these errors in a wide range of communication systems in computer memory, magnetic and optical data storage media, satellite and deep space communications, network communications, cellular telephone networks, and almost any other form of digital data communication. Digital data is transmitted over a channel and there is often noise in the channel. The noise may distort the messages to be sent. Therefore, what the receiver receives may not be the same as what the sender sends. The goal of error coding is to improve the reliability of digital communication by error detection and error correction [1].

A. Error

Data that is either transmitted over communication channel is not completely error free. This change in the data is caused due to external interference, signal distortion, attenuation or from noise. There are two types of errors. Firstly single error in which only one bit is changed. And secondly the burst error in which more than one bits are changed. There are various error detection and correction techniques such as Cyclic Redundancy Checks (CRC), Parity check, LRC, VRC and Hamming Code. This work focuses on Hamming code.

Manuscript published on 30 December 2014.
***** Correspondence Author (s)
Ravi Hosamani, Department of Electronics and Comminication Enginnering , VTU Belgaum, K.L.E. Institute of Technology, Hubli.
Ashwini S. Karne, Department of Electronics and Comminication Enginnering , VTU Belgaum, K.L.E. Institute of Technology, Hubli.

B. Hamming Code

A commonly known linear Block Code is the Hamming code. Hamming codes can detect and correct a single bit-error in a block of data. In these codes, every bit is included in a unique set of parity bits [2]. The presence and location of a single parity bit-error can be determined by analyzing parities of combinations of received bits to produce a table of parities each of which corresponds to a particular bit-error combination. This table of errors is known as the error syndrome. If all parities are correct according to this pattern, it can be concluded that there is not a single bit-error in the message (there may be multiple bit-errors). If there are errors in the parities caused by a single bit-error, the erroneous data bit can be found by adding up the positions of the erroneous parities. Hamming codes are easy to implement and are used in Hamming codes are generally used in computing, telecommunication, and other applications including data compression, and turbo codes [3]. They are also used for low cost and low power applications.

C. FPGA

A Field-Programmable Gate Array (FPGA) is a semiconductor device that can be configured by the customer or designer after manufacturing—hence the name "field-programmable". FPGAs are programmed using a source code in a Hardware Description Language (HDL) to specify how the chip will work [5].

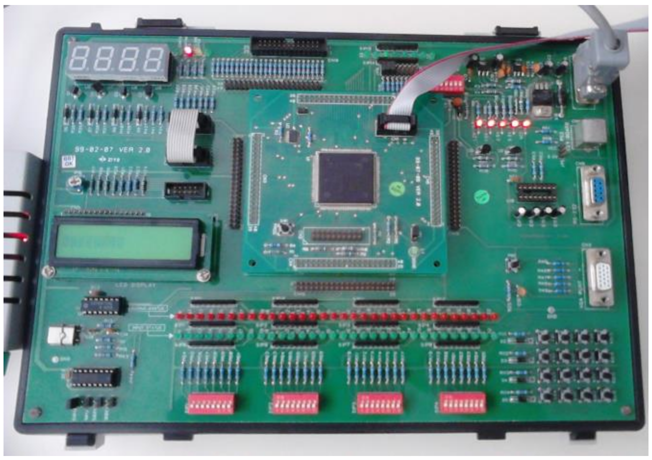

Figure 1: Spartan 3 FPGA trainer kit

The Spartan 3 FPGA is low-cost, high-performance logic solution for high-volume consumer-oriented applications. It also provides easy way to test the various programs in the FPGA itself, by dumping the 'bit' file into the FPGA and then observing the output. Fig 1. Shows Spartan 3 FPGA trainer kit device XC3S4O0-4PQ208 of Xilinx vendor.

The Spartan 3 FPGA board comes built in with many peripherals that help in the proper working of the board and also in interfacing the various signals to the board itself. Some of the peripherals included in the Spartan 3 FPGA board include are LCD, Keyboard port, VGA port, Two 9-pin RS-232 ports, 50 MHz clock oscillator, On board USB-based FPGA download and debug interface and 32 input/output pins for user interface. The Verilog source code has been edited and synthesized using xilinx14.3 then simulated using ISim. Spartan-3 FPGA trainer kit for Xilinx has been used for downloading the design(.bit file) in to FPGA chip. The material in this paper are organized as follows in section II brief discussion of FPGA design flow, Hamming encoding and decoding with error detection and correction is given in section III. In section IV the circuit design & top level design for encoder and decoder systems are described. The simulation and implementation results are discussed. at the end a conclusion is given in section V

II. DESIGN FLOW FOR FPGA

The process of implementing a design on an FPGA can be broken down into several stages, loosely definable as design entry or capture, synthesis, and place and route. Figure 2, shows the design steps required for implementing a design on FPGA.

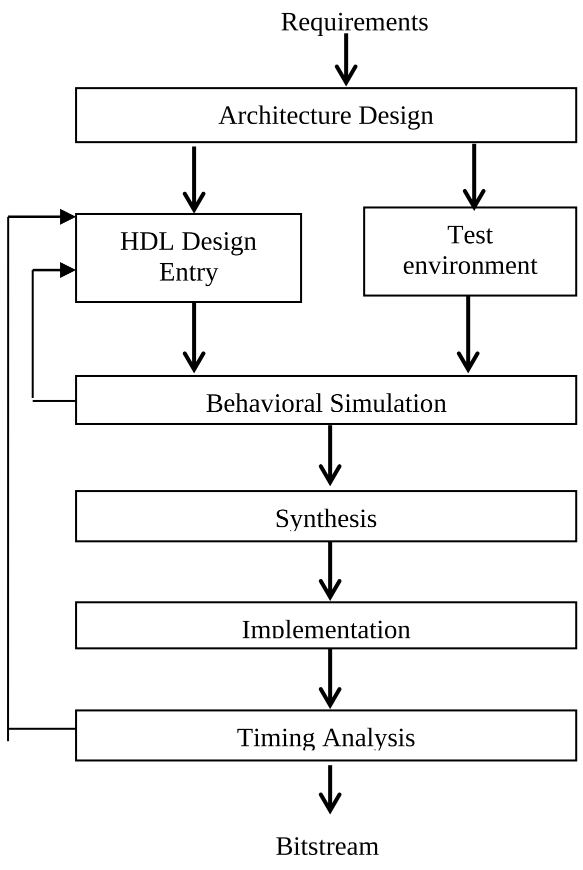

Figure 2: Schematic Showing design flow for FPGA

III. IMPLEMENTATION OF HAMMING CODE ON FPGA

A. ENCODING OF HAMMING CODE

To calculate the numbers of redundant bits (r) required to correct d data bits, let us find out the relationship between the two. So we have (d+r) as the total number of bits, which are to be transmitted; then r must be able to indicate at least d+r+1 different values. Of these, one value means no error, and remaining d+r values indicate error location of error in each of d+r locations. So, d+r+1 states must be distinguishable by r bits, and r bits can indicates 2^r states. Hence, 2^r must be greater than d+r+1. The value of r must be determined by putting in the value of d in the relation. For example, if d is 7, then the smallest value of r that satisfies the above relation is 4. So the total bits, which are to be transmitted is 11 bits (d + r = 7 + 4 =11). These redundancy bits are placed in positions 1, 2, 4 and 8 (the positions in an 11-bit sequence that are powers of 2). For clarity in the examples below, these bits are referred to as r1, r2, r3 and r4. In the Hamming code, each r bit is the parity bit for one combination of data bits as shown below:

r1: → 1, 3, 5, 7, 9, 11
r2: → 2, 3, 6, 7, 10, 11
r4: → 4, 5, 6, 7
r8: → 8, 9, 10, 11

11	10	9	8	7	6	5	4	3	2	1
d_6	d_5	d_4	r_8	d_3	d_2	d_1	r_4	d_0	r_2	r_1
Position of redundancy bits in Hamming Code										
Data:1010101										
11	10	9	8	7	6	5	4	3	2	1
1	0	1		0	1	0		1		
Adding r_1										
1	0	1		0	1	0		1		**1**
Adding r_2										
1	0	1		0	1	0		1	**1**	**1**
Adding r_4										
1	0	1		0	1	0	**1**	1	**1**	**1**
Adding r_8										
1	0	1	**0**	0	1	0	1	1	**1**	**1**
Encoded data:10100101111										

Figure 3: Redundancy bit calculation

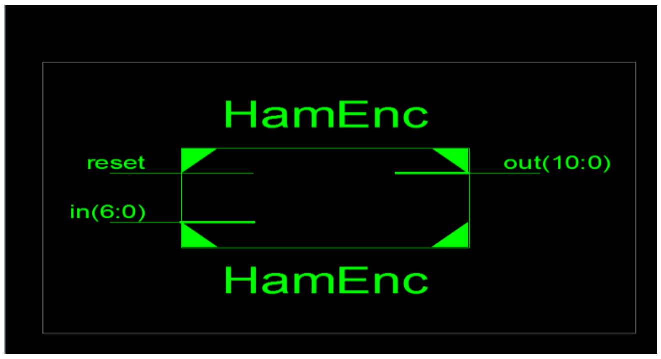

Figure 4: Top level of Hamming Encoder

The top level design of encode circuit is given in figure 4 shows encoder circuit with example of 7 bit information data with reset as inputs and 11 bit of encoded data with information bits and redundancy bit to be transmitted. Figure 5 shows RTL view of circuit shows main mux as main digital block of encoder and gate level view is show in figure 6.

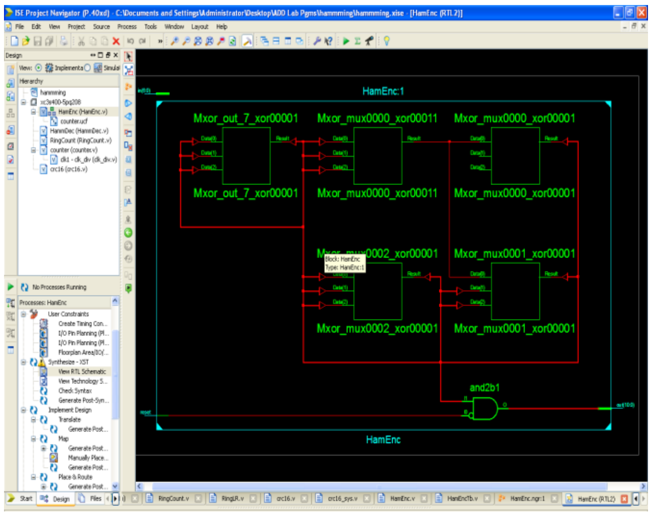

Figure 5: RTL view of Hamming encoder

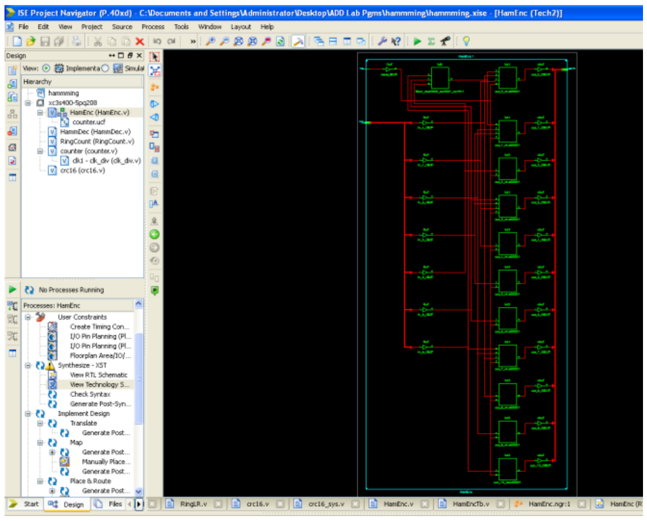

Figure 6: Technology view of Hamming encoder

B. DECODING, DETECTION AND CORRECTION OF HAMMING CODE

Table 1: Error position

Error position	Binary value of position			
0 (no error)	C2	C2	C1	C0
1	0	0	0	0
2	0	0	0	1
3	0	0	1	0
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1

At the receiving end the parity bits are recalculated. The decimal value of the k parity bits provides the bit-position in error, if any. The above table shows Hamming code is used for correction for 7-bit numbers (d_7 d_6 d_5 d_4 d_3 d_2 d_1) with the help of four redundant bits (r_4 r_3 r_2 r_1).with error positions (C3 C2 C1 C0) For the example data 100100101111, first r_1 is calculated considering the parity of the bit positions, 1, 3, 5, 7,9,11. Secondly the parity bits r_2 is calculated considering bit positions 2, 3, 6 7,10,11 then, the parity bits r_4 is calculated considering bit positions 4, 5, 6 and 7, finally the parity bits r_8 is calculated considering bit positions 8,9,10,11 as shown. If any corruption occurs in any of the transmitted code 10000101111, the bit position in error can be found out by calculating r_4 r_3 r_2 r_1 at the receiving end. For example, if the received code word is 1000010111, the recalculated value of r_4 r_3 r_2 r_1 is 1001, which indicates that bit position in error is 9, the decimal value of 1001[3].

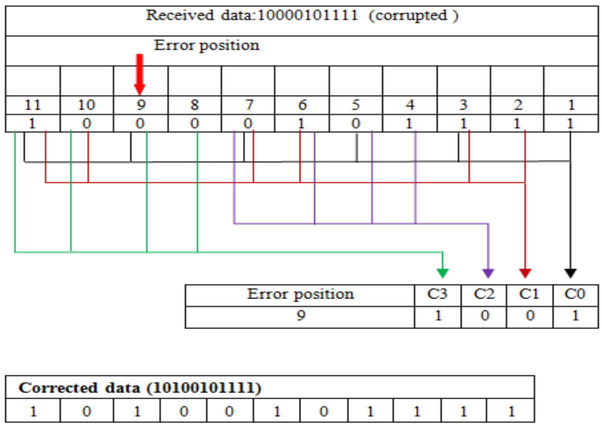

Figure 7: Error detection and correction using Hamming code

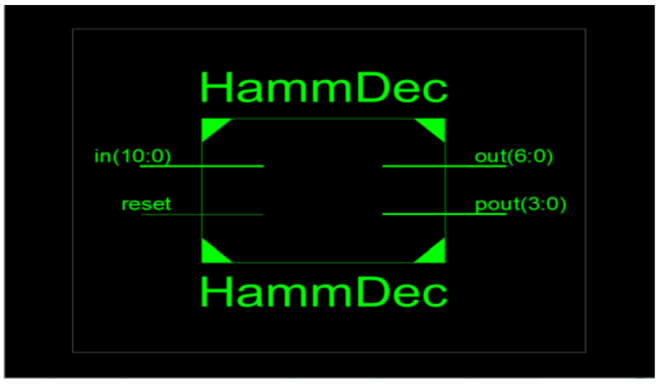

Figure 8: Top level of Hamming Decoder

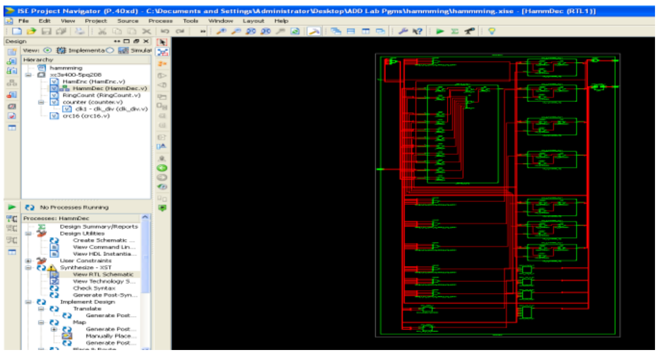

Figure 9: RTL view of Hamming Decoder

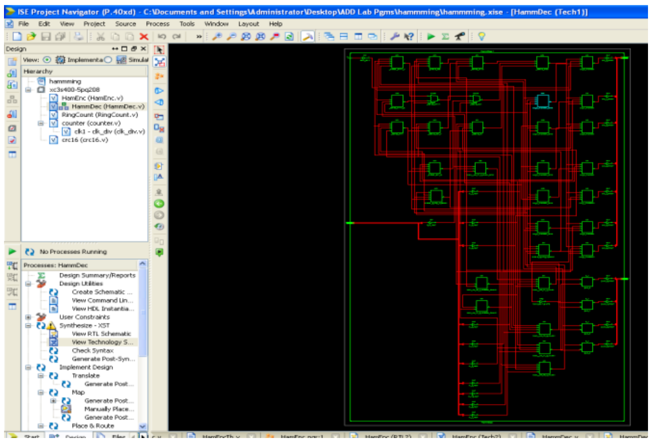

Figure 10: Technology view of Hamming Decoder

The top level design of Hamming Decoder circuit is shown in fig 8. Which shows decoder circuit with an example of 11 bit of data can be received with reset as inputs. 7 bits of output data which has error corrected information bits. The 4 bit of data as output to show the position of error if any. Figure 9 shows RTL view of circuit with main digital block and Figure10 gate level design of Hamming decoder system.

IV. RESULTS

A. Simulation Results

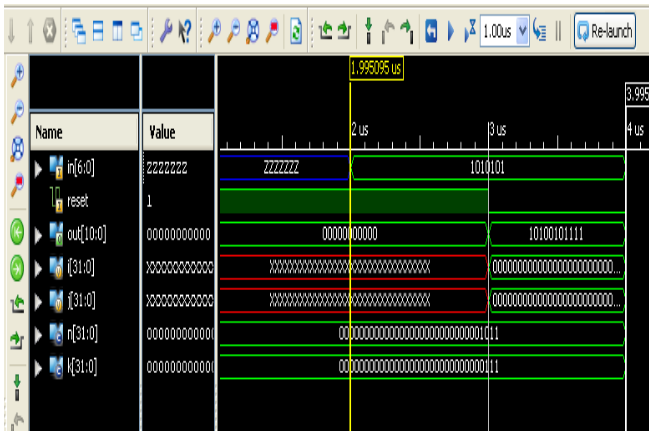

Figure 11: Simulation results of Hamming Encoder

The simulation Hamming Encoder result is shown in fig 11. Which is active low digital system with 7 bit as information bits as inputs in[6:0]. 11 bit output data as out[10:0] which has encoded data of encoder system . fig 11 show the example of input 7 bit data 1010101 as input and 10100101111 is the 11 bit encoded data.

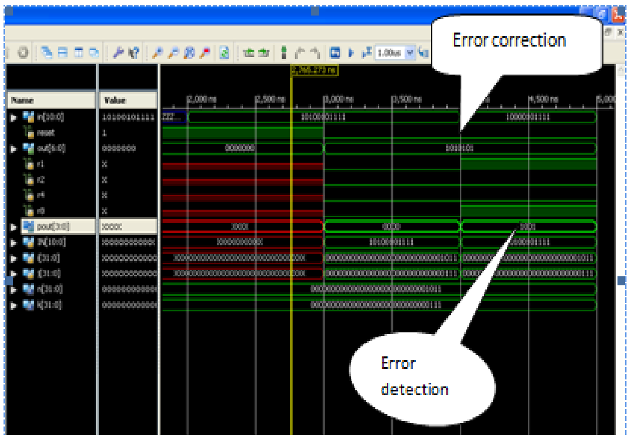

Figure 12: Simulation Results of Hamming Decoder

The simulation results of Hamming decoder is shown in Figure12 which show its active low digital system with 11 bit input data as in[10:0] and 7 bits of error corrected data in out[6:0] and another 4bit output data which holds the error bit position in received data pout[3:0]. Figure12 shows hamming decode with an example; received data is 10000101111 and the output obtained from the decoder data is 1010101, the place of error in the input data is shown as 1001.

B. Implementation Results

Hamming code encoder is implemented on Spartan 3 FPGA trainer kit using Xilinx and Verilog HDL.

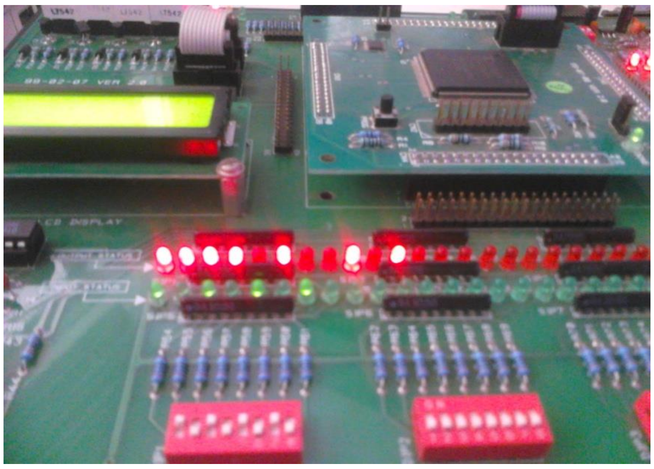

Figure 13: Implementation results of Hamming Encoder

Implementation results of Hamming Encoder on Spartan 3 FPGA trainer kit. The input are given using switches and ON(one) and OFF(zero) state of switch is indicate using green LED ON/OFF state(1010101)[4]. The output results shown with red LEDs binary ONE/ ZERO is indicated based on ON/OFF state of LED's The red LED's shows the encoded data 10100101111. Figure14 shows the implementation results of Hamming Decoder on Spartan 3 FPGA kit. The green LED with switch indicated the input(10000101111) to the decoder and Red LED's indicate the information data (1010101) and error position 1001.

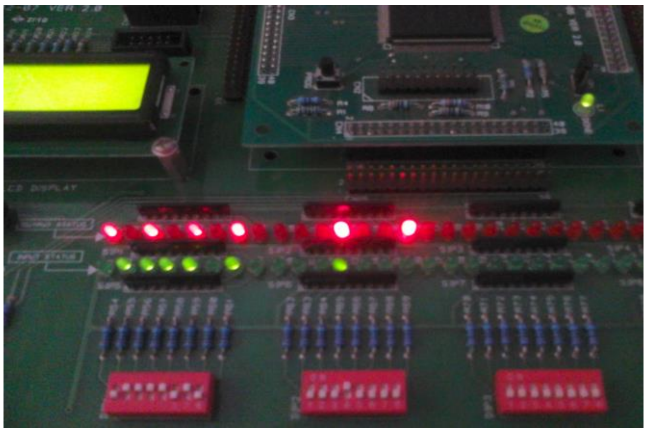

Figure 14: Implementation results of Hamming Decoder

V. CONCLUSION

FPGA Design of Hamming encoder and decoder with error detection and correction capabilities have been simulated and implemented.

The system has been designed using Verilog HDL and implemented on hardware using Xilinx 14.3, Spatan-3 FPGA starter kit. The Hamming encoding and decoding both are verified for different data input in the format (11,7,1) in both simulation and implementation in FPGA.

ACKNOWLEDGEMENT

The authors thank their parents and authorities of K.L.E. Institute of Technology, Gokul, Hubli, Karnataka, India for encouraging them for this research work.

REFERENCES

[1] Nuh Aydin: An Introduction to Coding Theory via Hamming Codes. Department of Mathematics Kenyon College.
[2] Ranpara, S.; Dong Sam Ha, 1999. A low-power Viterbi decoder design for wireless communications applications. IEEE Proceedings of the Twelfth Annual IEEE International Int. ASIC Conference 1999, Washington, DC, 15-18 Sept. 1999, pp. 377-381.
[3] Leena, Mr. Subham Gandhi and Mr. Jitender Khurana, “Implementing (7,4) Hamming Code using CPLD on VHDL” International Journal of New Trends in Electronics , Vol. 1, Issue 1, Aug. 2013.
[4] Xilinx “Spartan-3 FPGA Family, complete datasheet”, Xilinx Corp., Aug 2005.
[5] Xilinx “Synthesis and Simulation Design Guide”, Xilinx Tech UG626 2012.
[6] Ming- Bo Lin “ Digital System Design and Practices using Verilog HDL and FPGA”, Wiley-India , ISBN:978-81-265-3694-8 .
[7] Shu Lin, Daniel J. Costello. J “Error Control coding; Fundaments & applications”, Prentice Hall , ISBN 0-13-283796-X.1983

Mr. Ravi Hosamani, is working as Assistant Professor in Electronics and Communication Department at K.L.E institute of Technology, Hubli, Karnataka since 2010. He gets his M.Tech in VLSI design and Embedded systems from Visvesvaraya Technological University, Belgaum. His main research interests include Digital System Design in VLSI, Semiconductor device, Low Power VLSI Design, Embedded System and Wireless communication.

Miss. Ashwini S. Karne, is working as Assistant Professor in Electronics and Communication Department at K.L.E institute of Technology, Hubli, Karnataka since 2013. She gets her M.Tech in Digital Electronics from Visvesvaraya Technological University, Belgaum. Her main research interests include Embedded System and VLSI Design.