FPGA IMPLEMENTATION OF SOFT OUTPUT VITERBI ALGORITHM USING MEMORYLESS HYBRID REGISTER EXCHANGE METHOD

R .D. Kadam[1] and S. L. Haridas[2]

[1]Department of Electronics and Telecomm. BDCE, Sevagram, RTM Nagpur University, India
rdk_arvi@rediffmail.com
2Department of Electronics and Telecomm. BDCE, Sevagram, RTM Nagpur University, India
slhlec@rediffmail.com

ABSTRACT

The importance of convolutional codes is well established. They are widely used to encode digital data before transmission through noisy or error-prone communication channels to reduce occurrence of errors and memory. This paper presents novel decoding technique, memoryless Hybrid Register Exchange with simulation and FPGA implementation results. It requires single register as compared to Register Exchange Method (REM) & Hybrid Register Exchange Method (HREM); therefore the data trans-fer operations and ultimately the switching activity will get reduced.

KEYWORDS

Traceback method, Register Exchange Method, Hybrid Register Exchange Method, Memoryless HREM.

1. INTRODUCTION

The task facing the designer of a digital communication system is that of providing transmitting information from one end of the system at a rate and a level of reliability and quality that are acceptable to the user at the other end. Convolutional encoding with Viterbi decoding is a technique that is particularly suited to a channel in which the transmitted signal is corrupted mainly by additive white Gaussian noise (AWGN). Viterbi decoding was developed by Andrew J. Viterbi in 1967 [1]. As it was recognized by early 1970's, the algorithm was a maximum likelihood decision device for any symbol sequence that could be modeled as a state diagram [2]. Then in 1971 viterbi published another paper [3] focusing on convolutional codes, it begins with an elementary presentation of fundamental properties and structure of convolutional codes and proceeds with the development of maximum likelihood decoder. Since then, other researchers have expanded on his work by finding good convolutional codes, exploring the performance limits of the technique [4] and varying decoder design parameters to optimize the implementation of the technique for hardware and software. They are widely used to encode digital data before transmission through noisy or error prone communication channels to reduce occurrence of error. The traceback (other is register exchange) method is used as a data decoding technique offers reduced hardware complexity with longer latency as the trades off cyper et al [5] first present a algorithm for implementing a traceback survivor memory unit (SMU), which turned to be a generalization of the implementation is used in [6]. The hardware complexity of a viterbi decoder is proportional to the number of states in the trellis [7]. In order to make the Viterbi algorithm a practical decoding techniques certain refinement were made on the basis algorithm. The viterbi algorithm uses the trellis diagram to decode an input sequence [8] is a dynamic programming

DOI : 10.5121/vlsic.2011.2304

algorithm for finding the shortest path through a trellis. In case of soft decision technique, variation of the signal at the output of the demodulator are sampled and quantized, soft output viterbi algorithm accepts and delivers soft sample values and can be regarded as a device for improving the SNR [9], BMU finds the equivalent branch metric using correction between the received sample values and the expected symbols and add compared select (ACS) unit perform cumulative addition of branch metric to generate equivalent state metric, which multiple operation results in complex control logic. The two decoding methods called modified traceback and Hybrid register exchange method was proposed in [10]. A short overview of a soft input viterbi decoder implementation in field programmable gate array (FPGA) for code division multiple access (CDMA) wireless communication system is presented in [11]. It is well known that data transmission over wireless channels are affected by attenuation, distortion, interference and noise which affect the receiver's ability to receive correct information. Particularly for terrestrial cellular telephony, the interference suppression feature of CDMA can result in a many-fold increase in capacity with multiple access techniques were not as analog and even competing digital techniques [12].

In the following sections the coder used, system component, various decoding techniques and finally the proposed new decoding technique with simulation and FPGA results will be described.

2. CONVOLUTION CODES

Convolution codes are well described in the literature [1], [3]. They are commonly specified by three Parameters; (n, k, m), where: n is the number of output bits, k is the number of input bits, and m is the number of shift register stages of the coder. The constraint length K of the code represents the number of bits in the encoder memory that affect the generation of the n output bits and is defined as K = m + 1. The code rate r of the code is a measure of the code efficiency and is defined by r = k/n.

2.1. Structure of the convolutional code:

Figure 1 show the convolutional encoder structure (3, 1, 2) used in this paper and is built from its parameters. It consists of 2 (m=2) shift register stages and two modulo-3 adders (n = 3) giving the outputs of the encoder. The rate of the code is r = 1/3. The minimum distance of the code is d_{min} = 8. The outputs of the adders are sampled sequentially yielding the code symbols. The total number p of bit symbols is given by p = n (b + m) where b is the total number of bits of information [8].

The outputs Y_0, Y_1 and Y_2 of the adders are governed by the following generator polynomials. The generator polynomial for the output Y_0 and Y_1 is given by $g_0(x) = g_1(x) = 1 + x + x^2$ the generator polynomial for the output Y_2 is given by $g_2(x) = 1 + x^2$. $g_0(x)$, $g_1(x)$ and g_2 (x) select the shift register stages bits to be added to give the outputs of the encoder which are, for the case of (3, 1, 2) encoder as follow. The polynomials give the code its unique error protection quality.

$$Y2 = U0 \oplus U2 \qquad \text{----------------}1$$
$$Y1 = U0 \oplus U1 \oplus U2 \qquad \text{----------------}2$$
$$Y0 = U0 \oplus U1 \oplus U2 \qquad \text{----------------}3$$

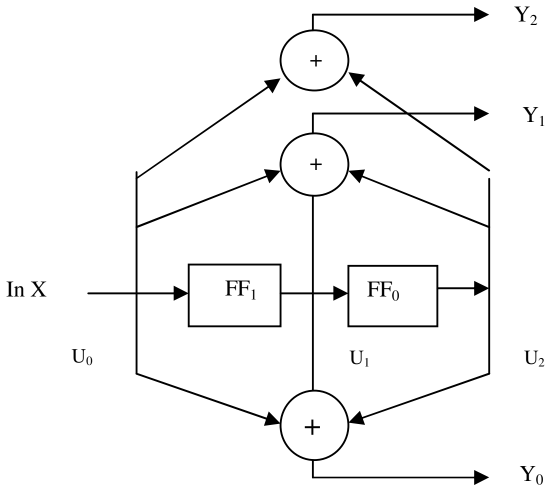

Figure 1. The 1/3 Convolutional Enc

3. Block Diagram of Viterbi Decoder

In this paper, the viterbi decoder is designed and implemented aims for 3GPP standard. The diagram of proposed architecture is shown in figure 2. It is composed of branch metric unit (BMU), add compare select unit (ACSU), survival memory unit (SMU), and a decoding unit.

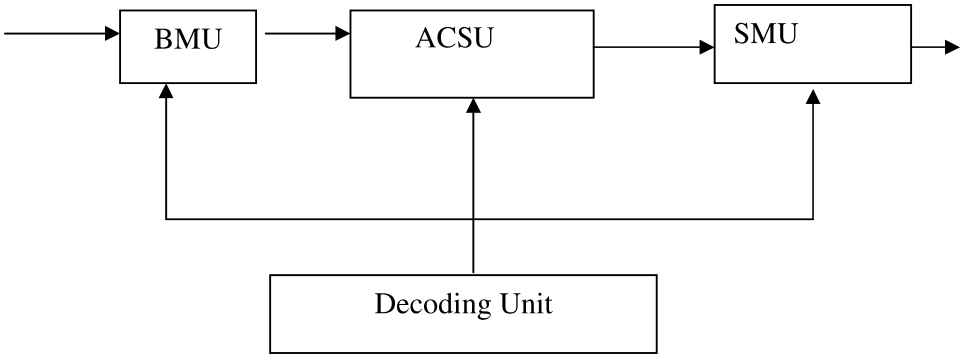

Figure 2. Block diagram of viterbi decoder

3.1. Branch Metric Unit (BMU)

BMU works for calculating the branch metrics according to received sequence. For three 3-bit soft decision input bits (i0, i1, i2) each ranging from -3 to +3, eight 5-bit branch metrics are generated. The decision bits are represented in the two's complement representation. The BMU perform simple add; subtract operations on the input bits to generate the output. For example the branch metric for the state transition which produces the binary output (010) is i0 - i1 + i2. The BMU performs the computations, as represented in table 1. The output of the BMU is still in a two's complement format. The bit serial format of the branch metrics is generated by the parallel to serial module at the output of the BMU, as shown in Figure 3. The bit serial format of the BMs is then fed into the ACSU. In the VA for decoding convolutional codes, the squared Euclidean distance is the optimum branch metric for decoding sequences that are transmitted in an AWGN channel. Multiplication operations or look up tables are required for the Viterbi algorithm to compute the squared distances to obtain the branch metrics. However, for binary convolutional codes, it is proven that linear distances (Hamming distances) can be used as the optimum branch metrics. This is true for convolutional codes.

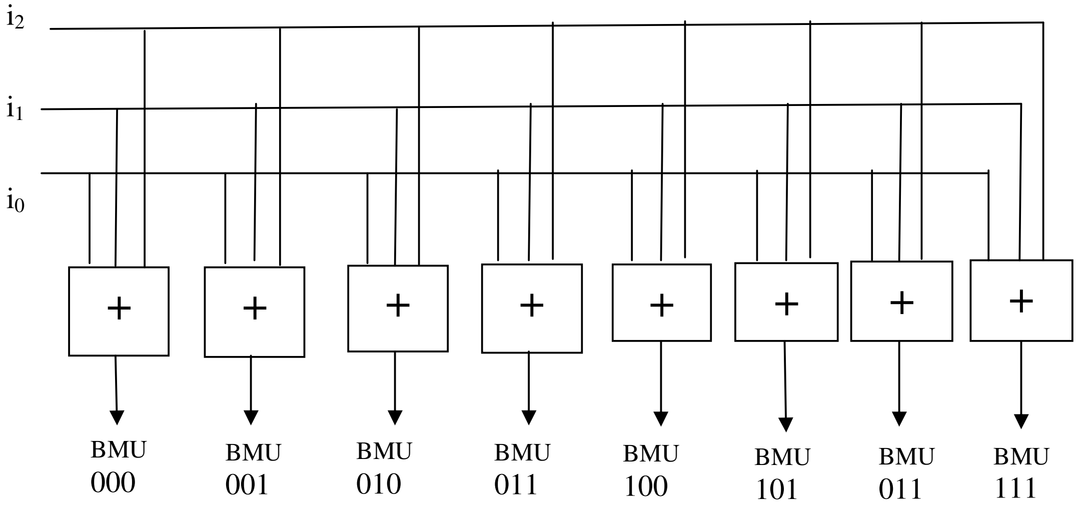

Figure 3. Branch Metric Unit

Table 1. Branch Metric

BMU000	$(i_0+i_1+i_2) = (-3-3-3)$	-9
BMU001	$(i_0+i_1-i_2) = (-3-3+3)$	+3
BMU010	$(i_0-i_1+i_2) = (-3+3-3)$	-3
BMU011	$(i_0-i_1-i_2) = (-3+3+3)$	+3
BMU100	$(-i_0+i_1+i_2) = (+3-3-3)$	-3
BMU101	$(-i_0+i_1-i_2) = (+3-3+3)$	+3
BMU110	$(-i_0-i_1+i_2) = (+3+3-3)$	+3
BMU111	$(-i_0-i_1-i_2) = (+3+3+3)$	+9

3.2. Add compare select (ACS)

The ACSU work in serial architecture style with 2 ACS as shown in figure 4, for having small implementation area, each trellis state S at time t is a state metric SMp and SMq which is the accumulated metric along the shortest path leading to that state.

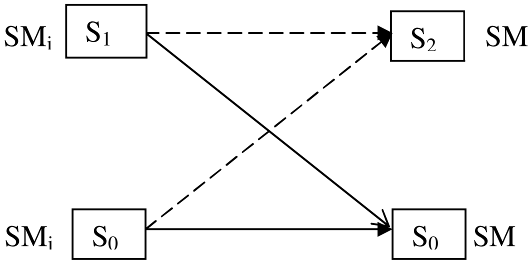

Figure 4. ACSU Butterfly

The state metrics at time can be recursively calculated in terms of the state metrics of the previous iteration as follows:

For state metric SMt

SMp = min (SMi+BMi1, SMj+BMj1),
SMq = min (SMi+BMi0, SMj+BMj0)

To demonstrate the functionality of VA, a sample input to the encoder is traced until the input is decoded. The encoder has an input sequence, 11011000 and generates the code stream, (111, 100, 100, 000, 100, 100, 111 and 000). The VA which uses soft decision formats to decode survivor path.

4. DECODING TECHNIQUES

4.1. Register Exchange method

In the register exchange method [2], a register assigned to each state contains information bits for the survivor path from the initial state to the current state. In fact, the register keeps the partially decoded output sequence along the path, as illustrated in figure 5. The register of state S_1 at t = 3 contain '110', which is the decoded output sequence along the hold path from the initial state. The resister exchange approach does require the copying of all the registers at each stage. The need to traceback is eliminated since the register of the final state contain the decoded output sequence. However, this approach results in complex hardware due to the need to copy the contents of all the register in a stage to the next stage. At last stage, the decoded output sequence is the one that is stored in the survivor path register S0, the register assigned to the state with the minimum Path Metric. Since the Register Exchange method does not need tracing back, it is faster. However, the RE method does require the copying of all the registers at each stage.

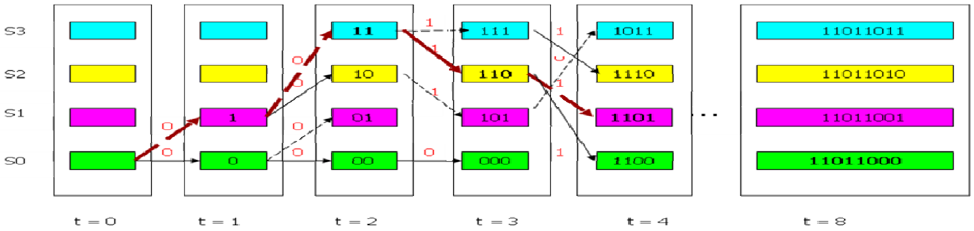

Figure 5. Register Exchange Approach

4.2. Hybrid Register Exchange Method

The main drawback of register exchange method is its frequent switching activity and long constraint length. One of the promising solutions to reduce the switching activity can be achieved by combining the REM and TB techniques [4]. The initial state can be first traced back through an m cycle, and then transfer the content of initial state to the current state and the next m bits of the register is the m bits of current state itself.

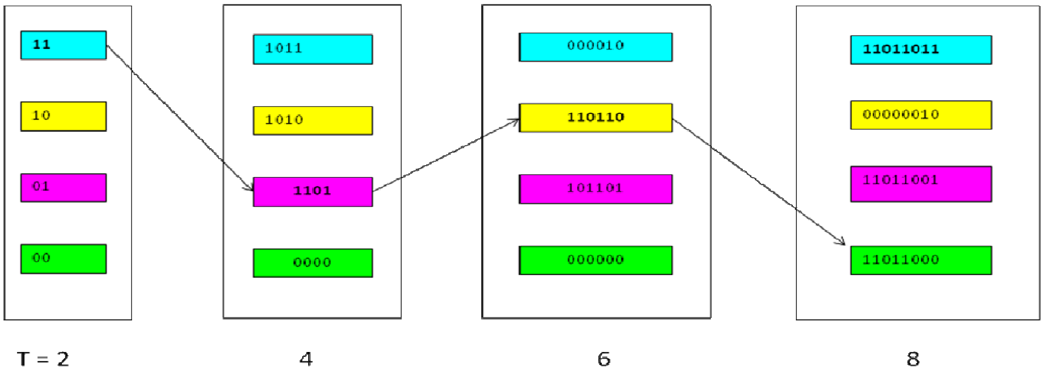

Figure 6. Hybrid Register Exchange Approach

4.3. Memoryless Register Exchange Method

The RE approach generates the decoded bits in the correct order. The decoded bits are produced, and then read out from the decoder. Thus, a memory free viterbi decoder can be implemented by solely resetting the encoder contents for each L bits that are encoded. The new VD implementation is called the memoryless viterbi decoder (MLVD). Since the MLVD needs to track only one row, the MLVD requires only one pointer to track the current position of the decoder in the trellis. If the initial state is zero, then only the first row of memory is needed. In other words, the storage of the decoded bits is necessary in order to choose only one row of memory at the end to represent the actual decoded bits. If the required row of memory is predetermined, and then there is no need for the storage of the other rows as shown in figure 7.

00	01	11	10	01	11	10	00	00
t=0	1 → t=1	11→ t=2	110→ t=3	1101→ t=4	11011→ t=5	110110→ t=6	1101100→ t=7	11011000 t=8

Figure 7. Memoryless REM

4.4. Memoryless Hybrid Register Exchange method

The MLVD keeps track of the current state position of the decoder in the memory unit as shown in Figure 8. It makes use of the fact that the bit appended to each row of memory is exactly the bit that is shifted into the pointer to form the new pointer to that row of memory [13]. To show the functionality, K = 3 a (four states) and rate = 1/3 convolutional encoder (g_2 = 101, g_1 = 111, g_0 = 111) is employed to encode the input sequence of (11011000). The code stream (111, 100, 100, 000, 100, 100, 111 and 000) is generated and transmitted over a channel. The noisy code stream (111, 110, 101, 000, 100, 100, 111 and 000) for example is received at the decoder. The underlined bits are incorrect because of the noise encountered during transmission. Applying the MLVD method result in the successive values for the pointer and row of memory for the decoded data over time. The pointer contains the current state of the decoder (m bits). The data at memory is the pointer value at that instant. Every time the digit get decode. It is reset to zero (the initial state of the encoder) after decoded last bits. After every 'm' cycles the pointer content and memory contents are upend, so the switching will get reduced.

11	01	10	00
→ 11 t=2	→ 1101 t=4	→ 110110 t=6	→ 11011000 t=8

Figure 8. Memoryless Hybrid Register Exchange Method

5. Simulation Result for Memoryless HREM

Figure 9 shows the simulation result for memoryless HREM, using pointer implementation. Encoding the data of 8 bit as adding 2 bit noise error. We get the result after 14 clock pulse unit as shown in Figure. It is same decoding input data to decode the serially output data. It shows the noise of two bits can be corrected by the decoder since it has free distance of 8.

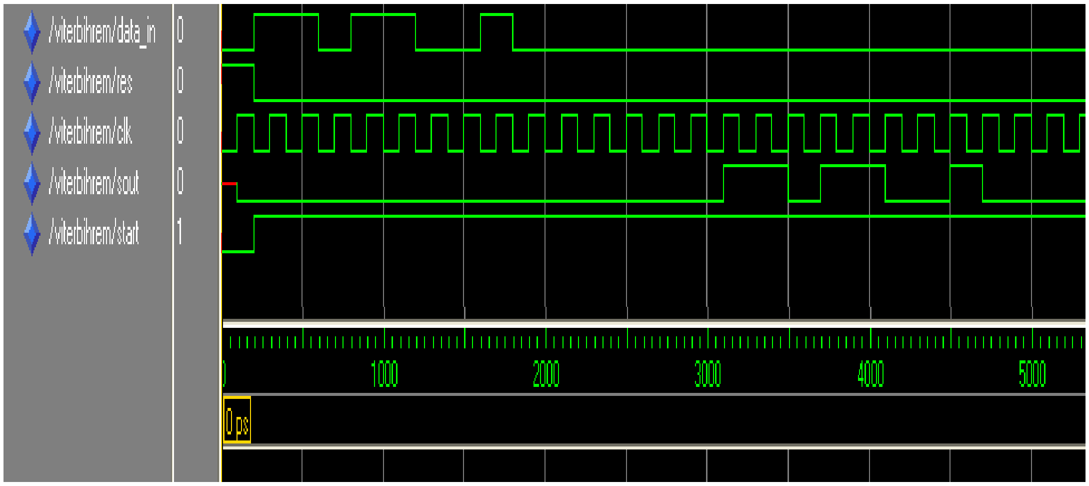

Fig.9. Simulation Result for memoryless hybrid register exchange method

6. FPGA Implementation

To prepare the MLVD using Hybrid Register Exchange processing VHDL design that is implemented on the FPGA, the design is synthesized by using Synopsys tool. Then, the design is imported into Xilinx tools for the mapping routing, then a VHDL file with timing information is

re-simulated for the timing verification. Afterward, the MLVDs design is downloaded to a xs2s200 Xilinx chip. The MLVDs consumes only 2% of the total slices of the xs2s200, comprising a total of 2352 gates. The design consumes 16 I/O bits.
All the decoding techniques are implemented and compared on the same platform.

Table 2. FPGA Simulation Report

Device Utilization	REM	HREM	MLREM	MLHREM
Number of Slices	254outof2352 10%	220 out of 2352 9%	60 out of 2352 2%	63 out of 2352 2%
Number of Slice Flip Flop	144outof4704 3%	142 out of 4704 3%	68 out of 4704 1%	68 out of 4704 1%
Number of 4 input LUTs	456outof4704 9%	390 out of 4704 8%	85 out of 4704 1%	83 out of 4704 1%
Number of bounded IOBs	16outof 144 11%	16 out of 144 11%	16outof 144 11%	16 out of 144 11%
Registers	87	85	47	47
Multiplexers	44	3	6	6
Adder/ Substractor	21	21	13	13
Comparators	9	9	3	3

7. CONCLUSION

The MLVD using HREM is a memoryless implementation of the VA, and successfully decodes the continuous data encoded by a convolutional encoder. The latency is only 2 data bits. The new implementation is realized by applying the pointer concept to the HREM implementation, and by using trellis truncation for every bit encoded. It is found that the new proposed decoding technique requires lesser hardware as compared to REM, HREM and memoryless REM techniques. The maximum throughput is 56 Mbps which is still more than 2Mbps which is requirement of wireless communication. Increasing the MLVDs performance by increasing the constraint length is still to be investigated. So the memoryless hybrid register exchange approach can be used in place of traceback, register exchange for decoding of data.

References

[1] A. J. Viterbi, “Error Bounds on Convolutional Codes and an Asymptotically Optimum Decoding Algorithm” Information and Control, 25(3), pp 260-269, April 1967.

[2] “A Personal History of the Viterbi algorithm” IEEE Signal Processing Magazine, July 2006.

[3] “Convolutional codes and their performance in communication systems” Andrew Viterbi, IEEE trans. On communication technology, vol. com-19, 1971.

[4] G. Forney, “Convolutional codes II. Maximum - likelihood decoding.” Information and Control, 25(3), pp 222-266, July 1974

[5] "Generalized Traceback Techniques for Survivor Memory Management in the Viterbi Algorithm", Robert Cypher, C. Bernard Shung, IEEE 1990.

[6] H.A. Bustamante, I. Kang, C. Nguyen and R.E. Peile," Stanford Telecomm design of a convolutional decoder," In MILCOM 89,Boston MA, October 1080 pp. 171-178.

[7] "A Low power Viterbi Decoder Design for wireless communication Application", Samirkumar Ranpara, Dong Sam Ha, IEEE 1999.

[8] A. J Viterrbi, J.K Omura, "Principals of digital communication", McGraw Hill, Inc, 1979.

[9] "A Viterbi Algorithm with soft decision outputs and its application", Joachim Hagenauer, Peter Hoeher, IEEE 1989.

[10] "Prof. S. L. Haridas, Dr. N. K. Choudhari," Design of Viterbi Decoder with Modified Traceback and Hybrid Register Exchange Processing ICAC3'09.

[11] "FPGA Implementation of soft Input Viterbi Decoder for CDMA2000 System" Milos Pilipovic, Marija Tadic, Novi Sad, Telefore 2008.

[12] "On the capacity of a cellular CDMA system "Klein S. Gilhousen, Irwin M. Jacobs, Roberto Padovani, Andrew Lindsay A. Weaver, Jr Charles E. Wheatley 1211, IEEE trans. On vehicular technology, vol.49, May 19991. "The CDMA Digital Cellular System an ASIC Overview" Richard Kerr, Kelein Gilhousen, Butch, Weaver, Robert Padovani, Houtan DSehesh, Qualcomm Inc, San Diego, IUEEE 1992.

[13] Memoryless Viterbi Decoder Dalia A. El-Dib, Member, IEEE and M. I. Elmasry, Fellow, EXPRESS BRIEFS, VOL. 52, NO. 12, IEEE 2005

Authors

Ravindra D. Kadam is pursuing M.Tech in Electronics Engineering from RTM Nagpur University, Nagpur, India. He received CDAC in Pune, India in 1999. He received his B.E. degree in Electronics Engineering from RTM Nagpur University, Nagpur in year 1997. He is currently working as Assistant Professor in the Department of Electronics and Telecomm. Engineering , BDCOE, Sevagram, Wardha, M.S., India. His research interests include computer networks and VLSI systems.

Sanjay L. Haridas received B.E. degree in Electronics Engineering from RTM Nagpur University, Nagpur in year 1988 and M.Tech in Electronics Engineering from VNIT, Nagpur in 1995. He is Professor in the Department of Electronics and Telecommunication Engineering, BDCOE, Sevagram, Wardha, M.S., India. His research interests are VLSI circuit design and Digital Communiation.