ijecs open access www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 4 Issue 8 Aug 2015, Page No. 13981-13986

Reconfigurable Modified Viterbi Decoder For A WiFi Receiver

Y.Radhika, G.Mukesh,M.Tech
Department of Electronics and communication Engineering
Audisankara College of Engineering & Technology, Gudur
(Autonomous)

Abstract : In every digital communication system, convolutional codes are the most efficient forward error correcting codes. Viterbi decoders are used to decode the convolutional codes. Together, Convolutional encoding with Viterbi decoding forms a powerful FEC technique when the message is corrupted by AWGN in a channel. General Viterbi Algorithm (VA), requires an exponential increase in hardware complexity to achieve greater decoder accuracy. When the decoding process uses the Modified Viterbi Algorithm (MVA), computations significantly gets reduced and results in the reduction of hardware utilization, which follows the maximum likelyhood path. In this paper, we present a Convolution Encoder and Viterbi Decoder with a constraint length of 3 and code rate of ½ and the results for hardware utilization of general VA and modified VA are compared. This is realized using Verilog HDL. The simulation and synthesis is done using Xilinx 14.3 ISE. The desin is implemented in FPGA VIRTEX kit.

Keywords: Convolutional Encoding, Viterbi Decoding, hardware reduction, maximum likelihood path, verilog HDL, FPGA.

I.INTRODUCTION

Convolution codes, which allow for efficient hard-decision[1]. It has been widely deployed in many wireless communication systems to improve the limited capacity of the communication channels. The Viterbi algorithm is the most extensively employed decoding algorithm for convolutional codes. The availability of wireless technology has revolutionized the way communication is done in our world today. With this increased availability comes increased dependence on the underlying systems to transmit information both quickly and accurately. Because the communication channels in wireless systems can be much more hostile than in “wired” systems, voice and data must use forward error correction coding to reduce the probability of channel effects corrupting the information being transmitted. Wi-Fi is a popular technology that allows an electronic device to exchange data wirelessly from transmitter to receiver.

A new type of coding, called Viterbi coding, can achieve a level of performance that comes closer to theoretical bounds than more conventional coding systems [2]. The Viterbi Algorithm, an application of dynamic programming, is widely used for estimation and detection problem in digital communications and signal processing. It is used to detect signals in communication channels with memory, and to decode sequential error control codes that are used to enhance the performance of digital communication systems [3]. Convolutional codes are frequently used to correct errors in noisy channels. They have rather good correcting capability and perform well even on very bad channels (with error probabilities of about (10□3) [4]. Convolutional codes are extensively used in satellite communications. Although convolutional encoding is a simple procedure, decoding of a convolutional code is much more complex task. Several classes of algorithms exist for this purpose.

Threshold decoding is the simplest of them, but it can be successfully applied only to the specific classes of convolutional codes. It is also far from optimal.

Sequential decoding is a class of algorithms performing much better than threshold algorithms. Their serious advantage is that decoding complexity is virtually independent from the length of the particular code. Although sequential algorithms are also suboptimal, they are successfully used with very long codes, where no other algorithm can be acceptable. The main drawback of sequential decoding is unpredictable decoding latency.

Viterbi decoding is an optimal (in a maximum-likelihood sense) algorithm for decoding of a convolutional code. Its main drawback is that the decoding complexity grows exponentially with the code length. So, it can be utilized only for relatively short codes.

Viterbi Decoding Techniques There are mainly two types of decoding techniques available in order to decode the data at the receiver end.

1) Register Exchange Method:
In this method, a register assigned to each state contains information bits for the survivor path from the initial state to current state. In fact, register keeps decoded output sequences along the path. This method requires copy of all registers at each stage. The need to trace back is eliminated since the register of final state contains decoded output sequence. This approach results in complex hardware due to need to copy contents of all register in a stage to next stage. Since the RE method does not need tracing back, it is faster.

2) Traceback Method

DOI: 10.18535/ijecs/v4i8.63

Trace back is memory organization method to store survivor paths and retrieve the decoded data. Direct implementation of this method is not practical because of an infinite storage size is needed; therefore in practice semiconductor infinite memory locations are reused periodically. The Trace Back Unit performs three operations: write new Data, Trace Back Read and Decode Read. Memory is organized as a two dimensional structure where row are assigned to states and columns to time steps. Three memory blocks are used in operation: write block is used to store ACS decision vectors, Decode block where the decoded bit sequences is read in backward order and Trace Back Block which is used to find the starting point of next trance back sequences. Traceback Depth (D) is a predefined parameter that defines the size of each memory block.To guarantee the convergence a traceback depth of D = 5K is sufficient and the memory block size will be 2K-1 x 5K. Traceback method is area efficient and better than RE method. . Register exchange method requires complex hardware compare to the Traceback method for larger constraint length though it will give faster speed.

In this paper , a hard decision and trace back method for viterbi decoding is implemented.

A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by a customer or a designer after manufacturing. FPGAs contain programmable logic components called "logic blocks", and a hierarchy of reconfigurable interconnects that allow the blocks to be "wired together “somewhat like many (changeable) logic gates that can be inter-wired in (many) distinct configurations. Complex combinational logic can be configured by using simple logic gates like AND and XOR. In most FPGAs, the sequential logic blocks include memory elements, which may be simple flip-flops or more complete blocks of memory.

II. CONVOLUTIONAL ENCODER

Convolutional Encoder shown in Fig.1 takes input data bit and gives out two bits. Convolutional encoding is a process of adding redundancy to a signal stream. It allows variable code rates (1/2), constraint lengths (K=3, 9) and generator polynomials [9], [10]. To convolve the encode data; start with 2 memory registers, each holding 1 input bit. Registers start with a value of 0. The encoder has 2 modulo-2 adders which are implemented with a XOR gate. It generates 2 bit polynomials, one for each adder. The convolutional encoder is basically a finite state machine. The k bit input is fed to the constraint length K shift register and the n outputs are calculated from the generator polynomials by the modulo-2 addition [10], [11]. The generator polynomial specifies the connections of the encoder to the modulo-2 adder. The ‘1’ in the generator polynomial indicates the connections and zero indicates no connections between the stage and modulo -2 adders.

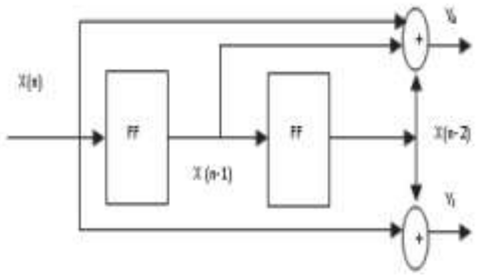

Figure 1: Rate = 1/2 encoder.
Y1Yo- Encoder output bits,
X (n-1) X (n-2)– previous state of Encoder,
X (n)-input bit to Encoder.

For a rate 1/2 encoder with constraint length of 3, the code can correct up to 2 errors in 16 bits of transmitted data. Here it is assumed that errors do not occur in consecutively. The code rate, is expressed as a ratio of the number of bits into the convolutional encoder (k) to the number of channel symbols output by the convolutional encoder (n) in a given encoder cycle. A rate 1/2 encoder is implemented in the design. The constraint length parameter, K, denotes the “length” of the convolution encoder, i.e. how many k-bit stages are available to feed the combinatorial logic that produces the output symbols. Closely related to K is the parameter m, which indicates how many encoder cycles an input bit is retained and used for encoding after it first appears at the input to the convolutional encoder. The following are the specifications of the encoder, K =3, Rate = ½ [12], [13]. Encoder functions depending on the applied input, and then the corresponding state transition takes place. The function of encoder understood with the help of the following state diagram. These state diagrams generally implemented with the sequential circuits based on the constraint length used at the transmitter side.

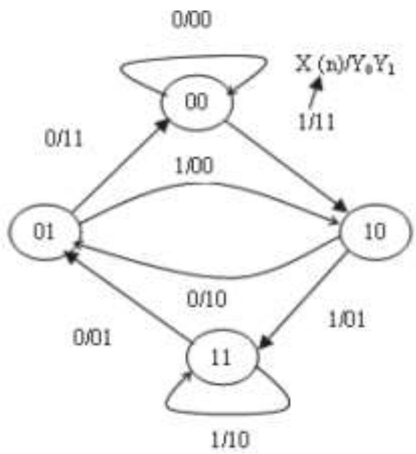

Figure 2: State diagram.

State diagram*:* This offers a complete description of the system. However, it shows only the instantaneous transitions. It does not illustrate how the states change in time.
Trellis diagram*:* To include time in state transitions [14].

II. VITERBI DECODER ARCHITECTURE

Figure 3: Viterbi decoder architecture.

The Fig. 1 shows that general Viterbi decoder architecture. This consists of three blocks
1)Branch Metric Unit (BMU)
2)Add Compare Select (ACSU) and
3)Survivor Memory Unit (SMU)
[5], [6], [7], [8].

1) Branch metric unit:

The first unit is called Branch metric unit. The Hamming distance (or other metric) values we compute at each time instant for the paths between the states at the previous time instant and the states at the current time instant are called branch metrics. Hamming distance or Euclidean distance is used for branch metric computation.

2) Path metric unit:

An accumulated Error metric called path metric (PM) contains the 2K-1 optimal paths. The current Branch Metric is added to previous PM and each the two distances are compared for all Add- compare select unit In terms of speed the performance of Viterbi Decoder is mainly determined by the number of ACS (2K-1) units and their computation time. As shown in figure each ACS unit comprises two adder blocks, a comparator and a selector block.

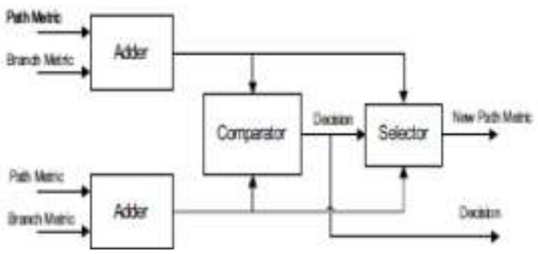

Figure 4: Block Diagram of Add Compare Select unit

3) Trace back:

This step is necessary for hardware implementations that don't store full information about the survivor paths, but store only one bit decision every time when one survivor path is selected from the two.

IV. VITERBI ALGORITHM

In the trellis diagram , horizontal direction circles shows the stages, vertical direction circles shows an ideal states and above which circles represents branch metric. Thick lines indicates encoding path for corresponding input data. 'T' indicates time slots for a clock.

Generally used the below trellis structure to decode the original data in the wireless communication receivers. The trellis used at the encoder and decoder same. But the hardware required for the decoding process is more, when constraint length becomes large[18], [19], [20], [21].

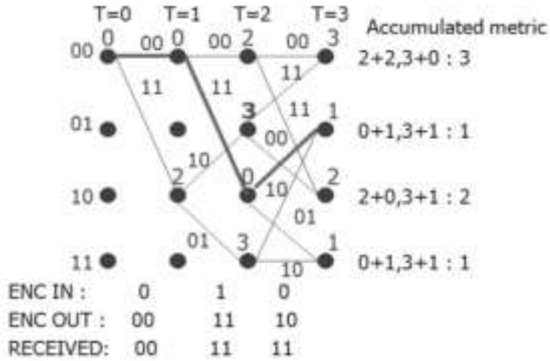

Figure 4. Trellis diagram.

V. MODIFIED VITERBI ALGORITHM

The average computation and path storage required by the MVA are reduced. Instead of computing and retaining all 2K-1 possible paths, only those paths which satisfy certain cost conditions are retained for each received symbol at each state node[1]. Path retention is based on the following criteria.

• A threshold 'T*'* indicates that a path is retained if its path metric is less than bm + T, where 'bm' is the minimum cost among all surviving paths in the previous trellis stage.

• The total number of survivor paths per trellis stage is limited to a fixed number, K, which is pre-set prior to the start of communication.

Path metrics are marked in bold on the nodes; dot lines indicate the least error path, upper branch indicates input bit '0', lower branch indicates input bit '1'.Output bits corresponding to the given input bit and state is shown on the branches.

In Fig. 5 Shows that constraint length k=3.In which the process can be continued with the help of threshold value. In this case fixed the threshold value as one, hence the process determination is not difficult, computations that are no of XOR operations performed at each stage reduced. The computations reduced drastically as constraint length increasing.

Generally the coding vectors used depending upon the code rate of encoder and constraint length associated with the input sequence. In the receiver side always consider the same process as transmitter, but in this process of decoding modified structure of decoder used. The following table shows the coding vectors associated with the corresponding constraint length.

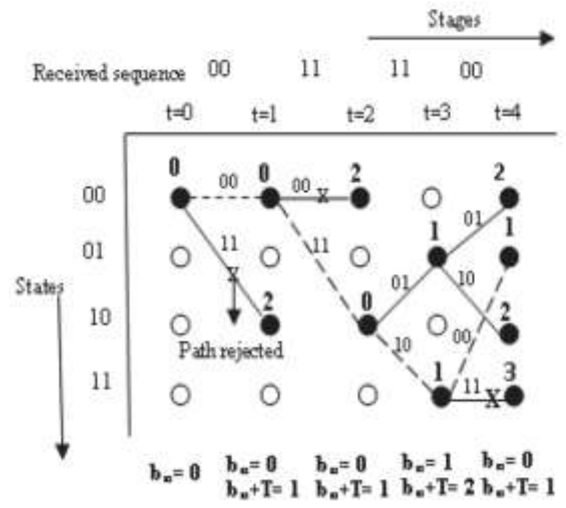

Figure 5: Trellis diagram for Modified Viterbi Algorithm

VI. EXPERIMENTAL RESULTS

In the evaluation of function of Viterbi decoder can be observed by manually introduce the errors in the received data. Then process the data into all the blocks with modified structure of decoder .Finally decoder decodes the original data by choosing the maximum likely path. Finally compare the original data and decoded data, if both are same decoder output shows the output as low. Because it specifies the errors present in the decoder output. This could be done by using 'XOR' operation between the original data and decoded data.

The memory specified by the Decoder depends on the addressing mode. If it is direct addressing mode then memory utilization associated with the decoder hardware less when compared to other addressing modes

The simulation waveforms and synthesis details of the encoding and viterbi decoding for the convolutional encoder specified above with k=3 and code rate=1/2 are as follows

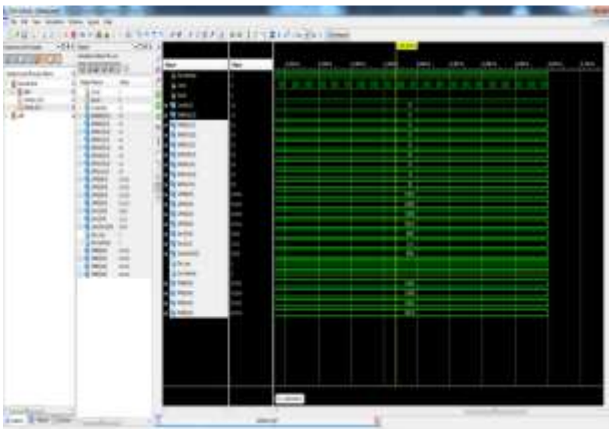

Figure 6: Simulation Waveform of Decoder

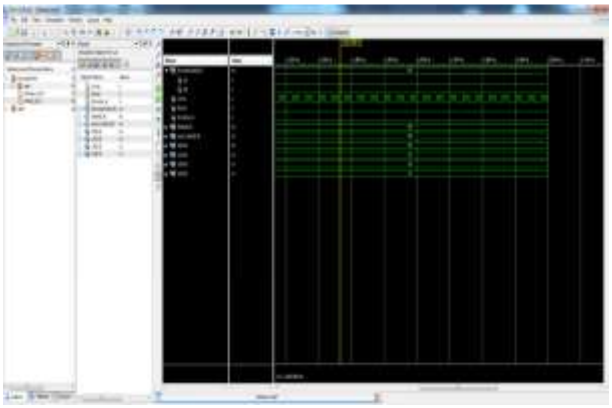

Figure 7: Simulation Waveform of Encoder

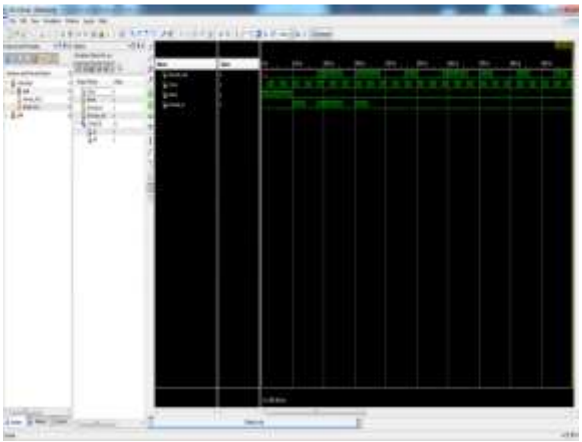

Figure 8: Overall Decoded Output for the encoded input Modified Viterbi Decoder(MVD)

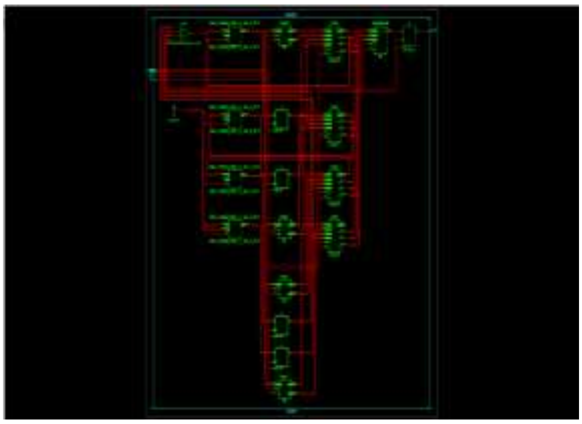

Figure 9: RTL Schematic for Decoder

DOI: 10.18535/ijecs/v4i8.63

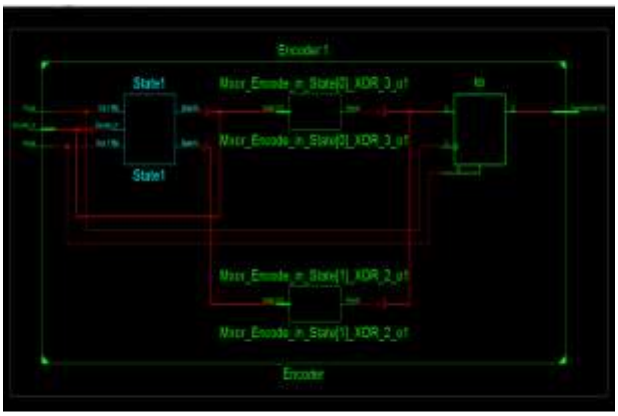

Figure 10: RTL Schematic for Encoder

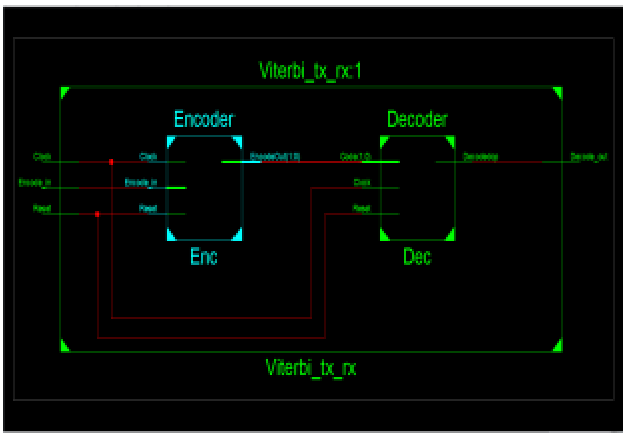

Figure 11: RTL Schematic for MVD

Figure 6,7 and 8 represents the simulation output waveforms for the Decoder Module, Encoder module and overall Modified Viterbi Decoder(MVD) respectively.

Figure 9,10 and 11 represents the Register Transfer Levels (RTLs) of Decoder Module, Encoder module and Modified Viterbi Decoder(MVD) respectively.

The hardware utilization for the Modified Viterbi Decoder with constraint length(k) of 3 and code rate(m) of ½ from the design synthesis report are as follows

Device Utilization summary for a conventional decoder (k=3)			
Logic Utilization	Used	Available	Utilisation
Number of slice registers	3	960	0%
Number of fully used LUT Flip-Flop pairs	4	1920	0%
Number of LUTs	6	1920	0%
Number of Bonded IOBs	11	66	16%
Number of BUFG/BUFGCTRLs	1	24	4%

Table 1: Hardware utilisation for a conventional decoder with k=3 and m=1/2

Device Utilization summary (k=3)			
Logic Utilization	Used	Available	Utilisation
Number of slice registers	38	42000	0%
Number of fully used LUT Flip-Flop pairs	79	21000	0%
Number of LUTs	33	84	39%
Number of Bonded IOBs	4	210	1%
Number of BUFG/BUFGCTRLs	1	32	3%

Table 2: Hardware utilisation for a MVD with K=3 and m=1/2

Figure 12: Hardware utilisation for MVD

Comparision of Table 1 and Table 2 clearly shows that the number of LUTs used has increased by 39% for a Modified Viterbi Decoder to that of a Conventional Decoder. The number of Bonded IOBs has tremendously reduced by 15% and the the number of BUFG/BUFGCTRLs has reduced by 1%.

VII.CONCLUSIONS

In this paper, modified Viterbi algorithm has been presented for a Wi-Fi receiver. The function of decoder with MVA determined through Xilinx ISE 14.3 tool in verilog. This approach has shown computations has significantly reduced which inturn has reduced the hardware requirement of overall decoder.

VIII.REFERENCES

[1] T. Kalavathidevi and C. Venkatesh, "Area Efficient Low Power VLSI Architecture for A Viterbi Decoder Using Gate Diffusion Input(GDI) Logic Style", European Journal of Scientific Research, Vol. 49, no.4, pp. 521-532, March 2011.

DOI: 10.18535/ijecs/v4i8.63

[2] Y. Shiau, P. Chen, H. Yang, Y. Lin and S. Huang, "An efficient VLSI architecture for convolution code decoding," International Symposium Next Generation Electronics, pp. 223-226, 2010.
[3] T. K. Trung, M. T. Shih, I. S Reed, and E. H. Satorius, "A VLSI design for a trace back viterbi decoder," IEEE Trans. on Communications, vol. 40, no.3, March 2009.
[4] S. Shaker, S. Elramly, and K. Shehata,"design and implementation of low-power viterbi decoder for software-defined WiMAX receiver," International Conference on Microelectronics, pp. 264 – 267, 2009.
[5] B. Pandita and S. K. Roy, "A Parallel Viterbi Decoder for Block Cyclic and Convolution Codes," Department of Electronics and Computer Science, University of Southampton, April 2008.
[6] C. Arun and P. Rajamani, "Design and VLSI implementation of a low probability of error viterbi decoder," First international conference on Emerging trends in Engineering and technology, pp. 418-423, 2008.
[7] S. Hema, V. Suresh babu and P. Ramesh, "Design and VLSI implementation of a low probability of error Viterbi decoder," First international conference on Emerging trends in Engineering and technology, pp.418-423, 2008.
[8] T. Gemmeke, V. S. Gierenz, and T. G. Noll, "RTL implementation of viterbi decoder," Dept. Of Computer Engineering, IEEE Transactions on circuits and systems, June 2006.
[9] V. Betz, "FPGAs and structured ASICs: Overview and research challenges". http://www.iic.umanitoba.ca/docs/vaughn-betz.ppt, October 2006.
[10] M. Kim and J. Cho, "A VLSI design of high speed bit-level viterbi decoder," IEEE transactions on Electrical & Electronics, pp.309-312, 2006.
[11] J. Man Guo and M. Omair Ahmad, "FPGA design and implementation of a low-power systolic array-based adaptive viterbi decoder," IEEE Transactions on circuits and systems, regular papers, Vol. 52, no.2, February 2005.
[12] F. Sun and T. Zhang, "Parallel gigh-throughput limited search trellis decoder VLSI design," IEEE Trans.Very Large Scale Integration System, vol. 13, no. 9, p. 1013-1022, 2005.
[13] M. Anders, S. Mathew, R. Krishnamurthy, and S. Borkar, "A 64-state 2GHz 500 Mbps 40 mW viterbi accelerator in 90 nm CMOS," in Symposium on VLSI Circuits, 2004.
[14] S. Swaminathan, R. Tessier, D. Goeckel, and W. Burleson, "A dynamically reconfigurable adaptive viterbi decoder," Monterey, Calofornia, USA, February 24-26, 2002.
[15] K. Chadha and J. R. Cavallaro, "A reconfigurable viterbi decoder architecture," Proceedings of Asilomar Conference on Signals, Systems and Computers, pp. 66–71, Nov. 2001.
[16] B. Sklar, Digital Communications - Fundamentals and Applications, Prentice Hall, New Jersey, 2001.
[17] J. Proakis. Digital Communications. McGraw-Hill, New York, NY, 1995.
[18] P. J. Black and T. H.Y. Meng, "Hybrid survivor path architectures for Viterbi decoders," Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 433–436, April 1993.
[19] P. J. Black and T. H.Y. Meng, "Hybrid survivor path architectures for viterbi decoders," Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 433–436, April 1993.
[20] S.J. Simmons, "Breath-first trellis decoding with adaptive effort," IEEE Transactions on Communications, vol. 38, pp. 3-12, January 1990.
[21] J. B. Cain, G. C. Clark, Jr., and J. M. Geist, "Punctured convolutional codes of rate (n 1)/n and simplified maximum likelihood decoding,"
IEEE Transactions on Information Theory, pp. 97–100, January 1979.

Author's Profile:

Y.Radhika received her B.Tech degree in Electronics and Communication Engineering from Narayana Engineering College, Gudur, S.P.S.R. Nellore District, affiliated to JNTU Anantapur. She is currently pursuing M.Tech VLSI in Audisankara college of Engineering and Technology, Gudur(Autonomous), SPSR Nellore (Dist), affiliated to JNTU Anantapur.

G.MUKESH working as Assistant Professor in ECE dept, ASCET (autonomous). He purused B.tech from Narayana Engineering college, Gudur and pursued M.tech from PBR VITS , Kavali.

He is guiding UG and PG students. His interested areas are Embedded System and VLSI.