International Journal of Engineering Research and General Science Volume 4, Issue 1, January-February, 2016
ISSN 2091-2730

Simulation of Convolution Encoder and Viterbi decoder Using Verilog HDL

Rakhi B. Menon[1], Dr. Gnana Sheela K[2]

M.Tech Student[1], Professor[2], Department of ECE, TOCH Institute of Science and Technology, Arakkunnam

Abstract— Convolution encoder and Viterbi decoder of rate 2/3 is simulated using Model Sim. Test benches for encoder and decoder are coded and simulated. Two bits are provided to the encoder as input and three bits are obtained as output of encoder. These encoded three bits are provided as input to the Viterbi decoder. Viterbi decoder consists of different modules such as Euclidean distance calculation module, Subset decode module, Compute metric module, Compare Select Module, Path module, Path memory module, Path in Module, Metric Module, Output decision module and reduce module. All these modules are internally connected by Verilog HDL codes and simulated using Model Sim. The output of the Viterbi decoder will be two bit output. When Input to the Convolution encoder is compared to the output of Viterbi decoder, results shows that they are the same. Convolution Encoder with Viterbi decoder finds applicable in digital broadcasting, Satellite applications, Digital Mobile Applications, Deep Space, Code Division Multiple Access and voice-band data communications.

Keywords— Convolution encoder, Viterbi decoder, Viterbi Algorithm, Verilog HDL, Soft decision decoding, Trace back method, Model Sim

INTRODUCTION

Convolution codes were invented in 1955 by P.Elias. Convolution codes are generally error correcting codes that are used to improve the performance of many digital systems such as digital radio, mobile phones and the Bluetooth implementations. Viterbi decoder together with its improved versions is one of the best applications of convolutional codes. Convolutional coding has been used in communication systems including deep space communications and wireless communications. It offers an alternative to block codes for transmission over a noisy channel. An advantage of convolutional coding is that it can be applied to a continuous data stream as well as to blocks of data. Convolution codes are also used in real time error correction to improve the performance of digital radio, mobile phones, satellite links etc. The simplicity and performance of convolution codes for Gaussian channel is very close to the accurate. They are one of the most widely used channel codes in the practical communication systems.

There are three alternative methods that are often used to describe the convolutional code. These are the tree diagram, state diagram and trellis diagram. There exist four basic convolutional codes decoding techniques: sequential, threshold, maximal-likelihood and the Viterbi algorithm. The sequential algorithm can provide very strong correcting capabilities while it needs relatively large memory, which strongly depends on communication channel error density. The threshold algorithm is extensively good for channels with mid to good signal to noise ratios (SNR). The Viterbi algorithm is an optimum decoding technique. It is optimum as it results in the minimum probability of error. It is also the relatively straight algorithm to implement in hardware and is the best decoding technique. Viterbi algorithm is a maximum likelihood algorithm and performs decoding, through searching the minimum cost path in a weighted oriented graph, called trellis. The basic building blocks of Viterbi decoder are branch metric unit (BMU), path metric unit (PMU), add compare and select unit (ACSU) and survivor memory management unit (SMU).

CONVOLUTION ENCODER

Convolution encoding is widely used for satellite and other noisy communications channels. There are two important components of a channel using Convolution encoding: the Convolution encoder (at the transmitter) and the Viterbi decoder (at the receiver). An encoder includes extra information in the transmitted signal to reduce the probability of errors in the received signal that may be corrupted by noise. Every two bits of data stream are encoded into three bits for transmission. The ratio of input to output information in an encoder is the rate of the encoder; this is a rate 2/3 encoder. The following equations relate the three encoder output bits (Yn2 , Yn1 , and Yn0) to the two encoder input bits (Xn1 and Xn0) at a time nT:

$$Yn2 = Xn1 \quad (1.1)$$
$$Yn1 = Xn0 \text{ xor } Df2 \quad (1.2)$$
$$Yn0 = Df1 \quad (1.3)$$

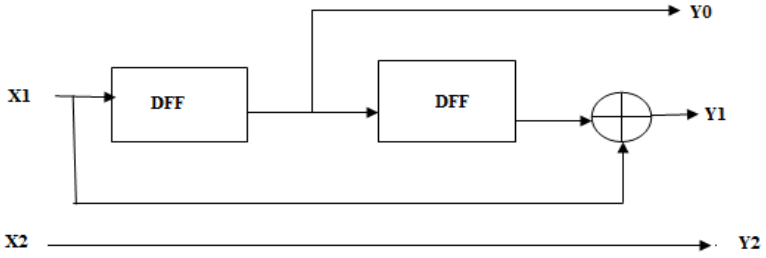

Figure1. Convolution encoder of rate 2/3

VITERBI DECODER

The Viterbi decoder consists of three main units branch matrices unit, add control select unit and survivor management unit. The branch metric unit (BMU) is responsible for the computation of matrices, second block add compare select unit (ACSU) selects the survivor paths for each trellis state, third block survivor management unit (SMU) performs the selection of output which is based on the most optimum path metric. Figure shown below shows the general structure of Viterbi decoder

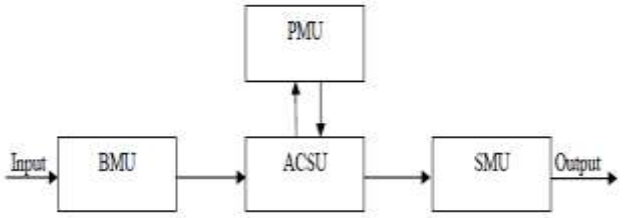

Figure2. Block diagram of Viterbi decoder

INTERNAL MODULES OF VITERBI DECODER

The Viterbi decoder consists of various internal modules for Branch metric calculation, Path metric calculation, modules to perform addition of branch metric and path metric, comparison module to select the smallest path and other modules. The first module is Euclidean distance calculation module. The other modules are subset decode module, Compute metric module, Compare Select module, Path in and Path memory modules, Reduce metric and Output decision module.

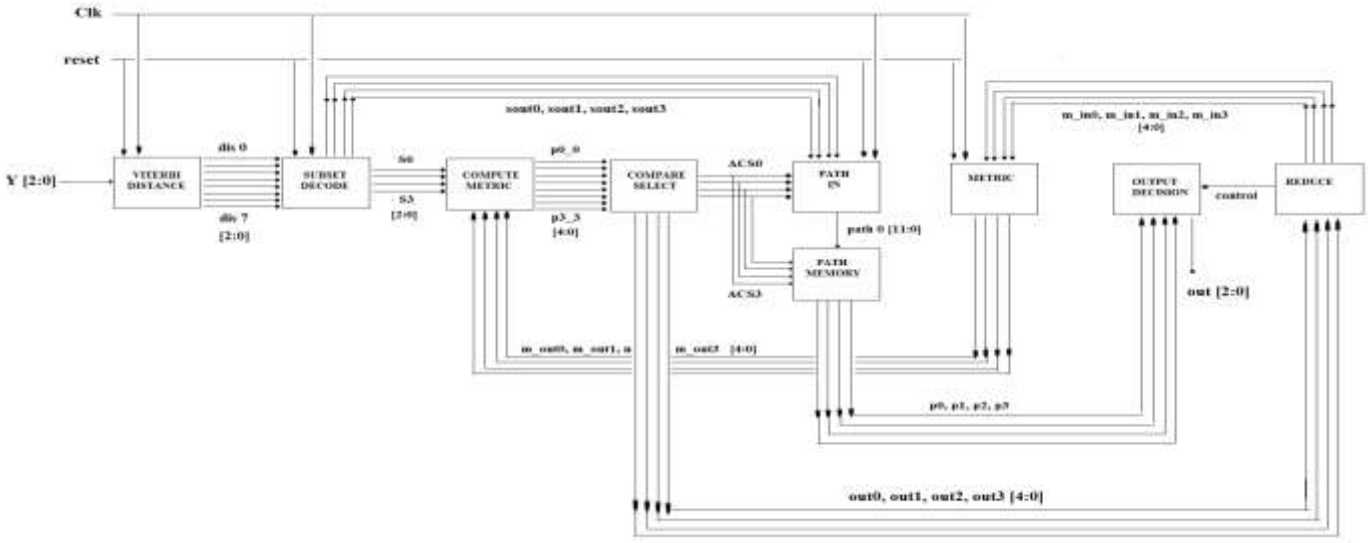

Figure3. Internal modules of Viterbi decoder

Viterbi distance

The received signal (with noise) is converted into a series of distance measures from the known eight possible transmitted signals: The digitally encoded 3-bit signal, Y, from the encoder is converted directly to the distance measures. d[N] is the distance from signal = N to signal = 0 d[N] = (2*sin(N*PI/8))**2 in 3-bit binary (on the scale 2=100)

Table 1.1 Calculation of Euclidean distance

Signal	Algebraic distance from signal 0	X =Distance from signal 0	Euclidean distance $E = X^2$	B = binary quantized value of E	D = decimal value of B	Quantization error Q = D - 1.75 E
0	2 sin (0 π / 8)	0.00	0.00	000	0	0
1	2 sin (1 π / 8)	0.77	0.59	001	1	-0.0325
2	2 sin (2 π / 8)	1.41	2.00	100	4	0.5
3	2 sin (3 π / 8)	1.85	3.41	110	6	0.0325
4	2 sin (4 π / 8)	2.00	4.00	111	7	0
5	2 sin (5 π / 8)	1.85	3.41	110	6	0.0325
6	2 sin (6 π / 8)	1.41	2.00	100	4	0.5
7	2 sin (7 π / 8)	0.77	0.59	001	1	-0.0325

Module Subset decode

This module chooses the signal corresponding to the smallest of each set {||r-s0||**2,||r-s4||**2}, {||r-s1||**2, ||rs5||** 2}, {||r-s2||**2,||r-s6||**2}, {||r-s3||**2,||r-s7||**2}. Therefore there are eight input signals and four output signals for the distance measures. The signals sout0, sout3 are used to control the path memory. The statement dff #(3) instantiates a vector array of 3 D flip-flops.

Module Compute metric

This module computes the sum of path memory and the distance for each path entering a state of the trellis. For the four states, there are two paths entering it; therefore eight sums are computed in this module. The path metrics and output sums are 5 bits wide. The output sum is bounded and should never be greater than 5 bits for a valid input signal. The overflow from the sum is the error output and indicates an invalid input signal.

Module Compare select

This module compares the summations from the Compute metric module and selects the metric and path with the lowest value. The output of this module is saved as the new path metric for each state. The ACS output signals are used to control the path memory of the decoder.

Module path

This is the basic unit for the path memory of the Viterbi decoder. It consists of four 3-bit D flip-flops in parallel. There is a 2:1 mux at each D flip-flop input. The statement dff #(12) instantiates a vector array of 12 flip-flops.

Module Path memory

This module consists of an array of memory elements (D flip-flops) that store and shift the path memory as new signals are added to the four paths (or four most likely sequences of signals). These module instantiates 11 instances of the path module.

Module Path in

This module determines the input signal to the path for each of the four paths. Control signals from the subset decoder and compare select modules are used to store the correct signal. The statement dff #(12) instantiates a vector array of 12 flip-flops.

Module metric

The registers created in this module (using D flip-flops) store the four path metrics. Each register is 5 bits wide. The statement dff #(5) instantiates a vector array of 5 flip-flops.

Module Output decision

This module decides the output signal based on the path that corresponds to the smallest metric. The control signal comes from the reduce module.

Module reduce

This module reduces the metrics after the addition and compares operations. This algorithm selects the smallest metric and subtracts it from all the other metrics

SIMULATION RESULTS OF CONVOLUTION ENCODER

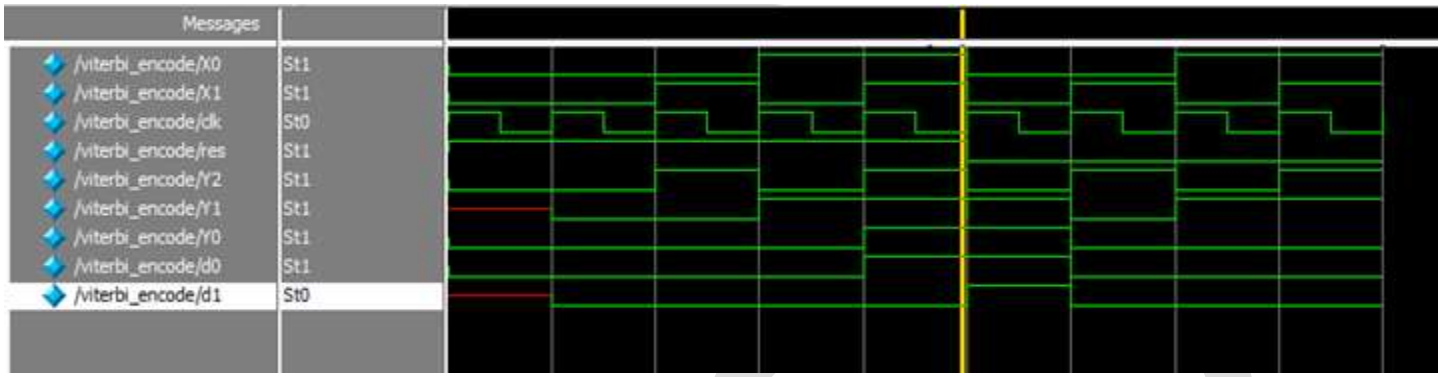

Figure4. Simulation of convolutional encoder

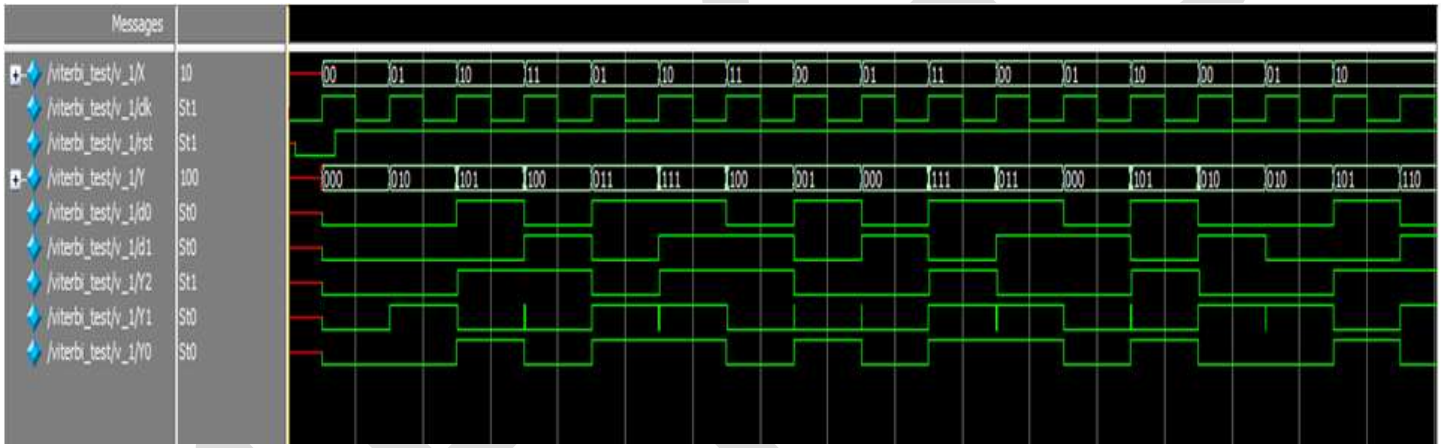

Figure5. Test bench of convolutional encoder

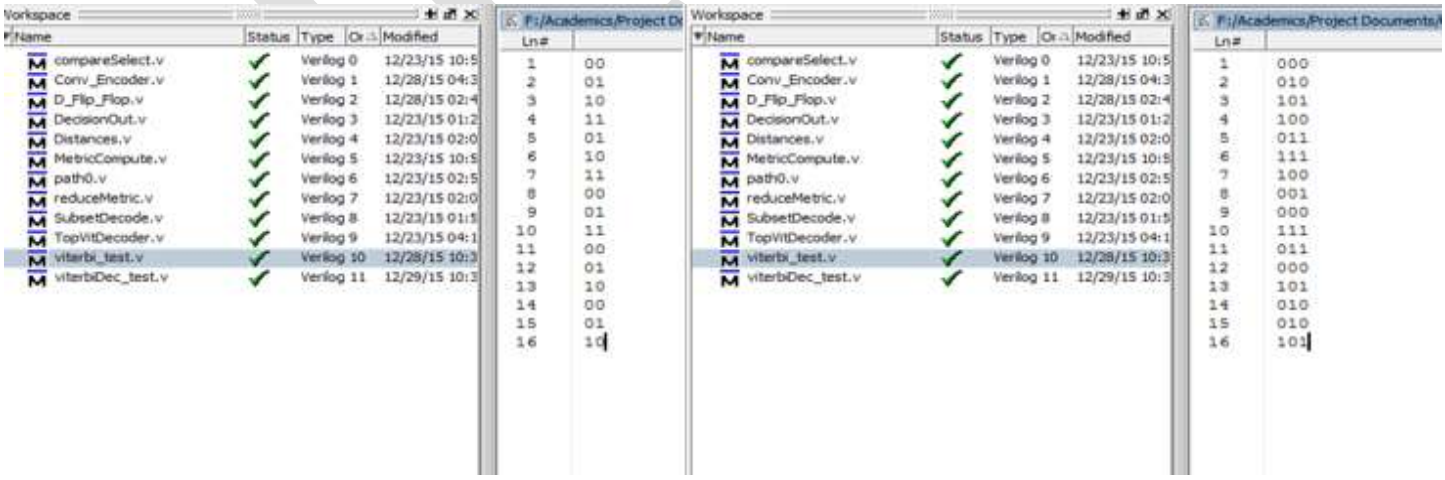

Figure6. Input and Output of Convolution encoder of rate 2/3

Two bit inputs are provided to Convolution encoder. The inputs are taken as text document. Thus the convolution encoder of rate 2/3 is simulated successfully and three bit outputs are obtained which confirms that the simulation is successful. Test bench results prove the same. Two bit input and three bit output are visible on the test bench.

SIMULATION RESULTS OF VITERBI DECODER

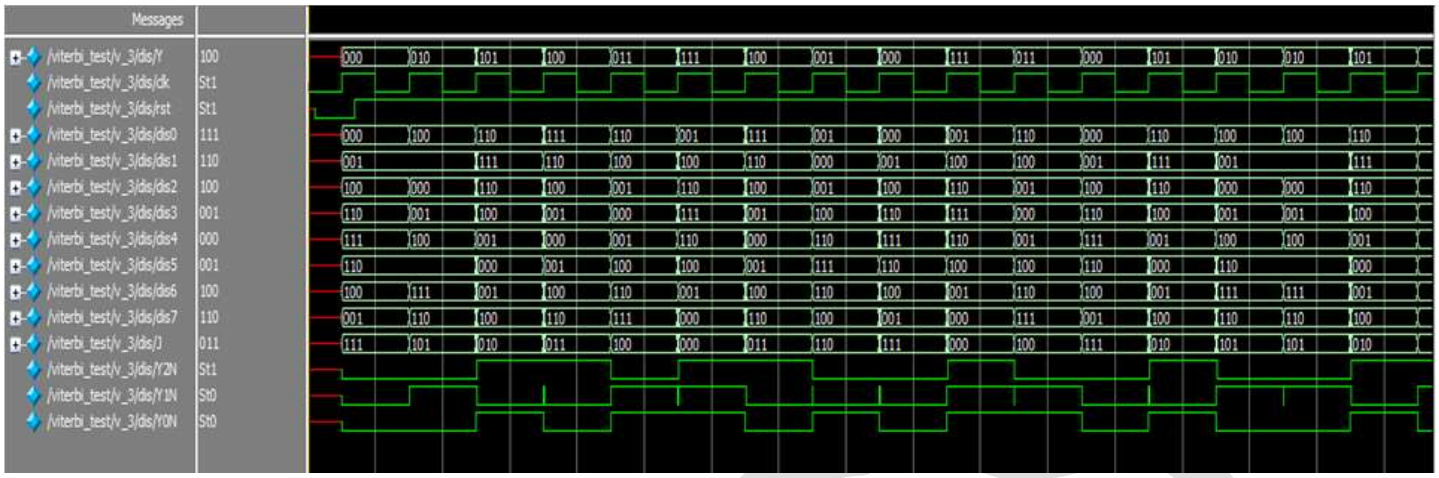

Figure7. Calculation of Euclidean distance

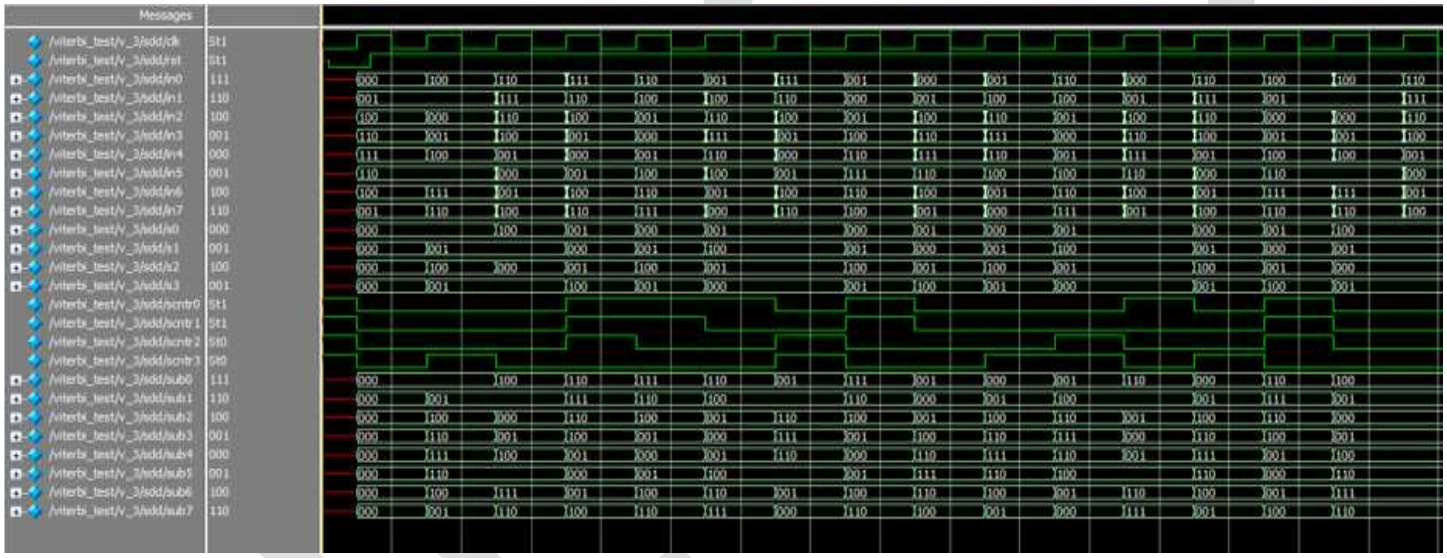

Figure8. Subset Decode Module

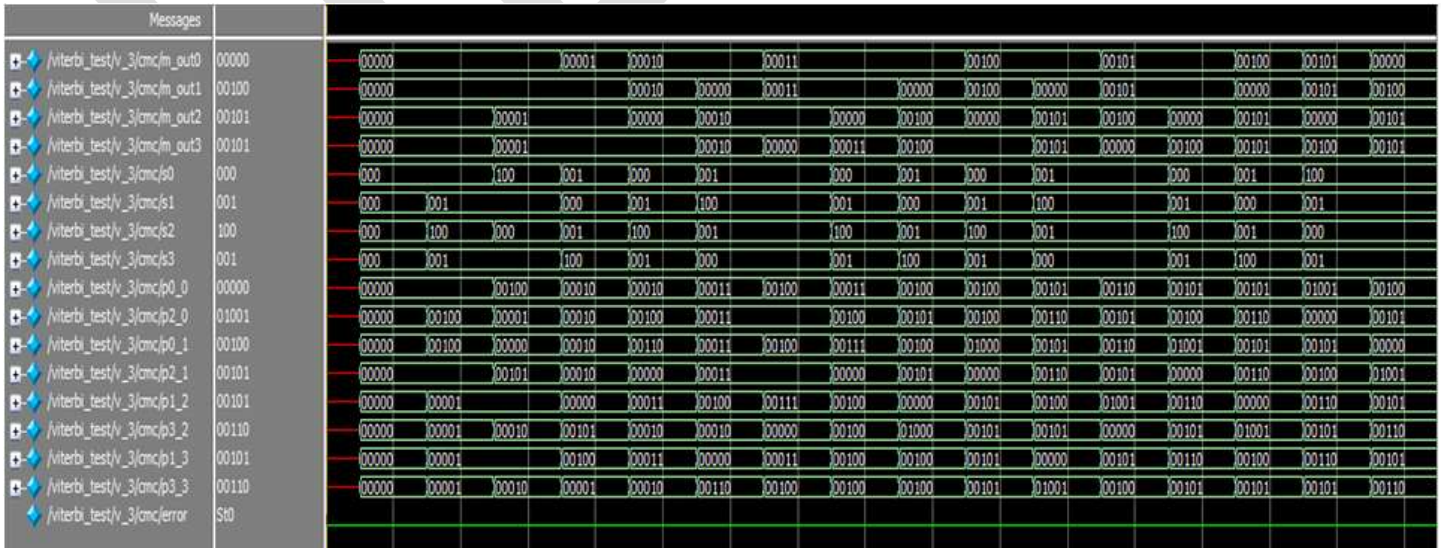

Figure9. Compute Metrics Module

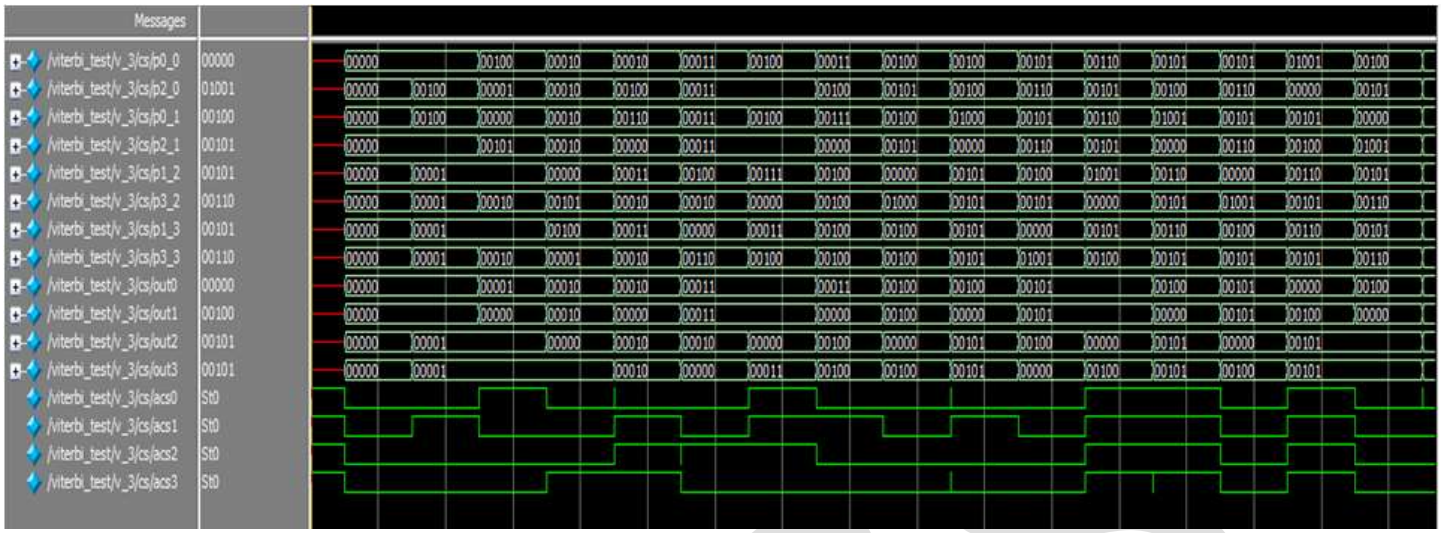

Figure10. Compare Select Module

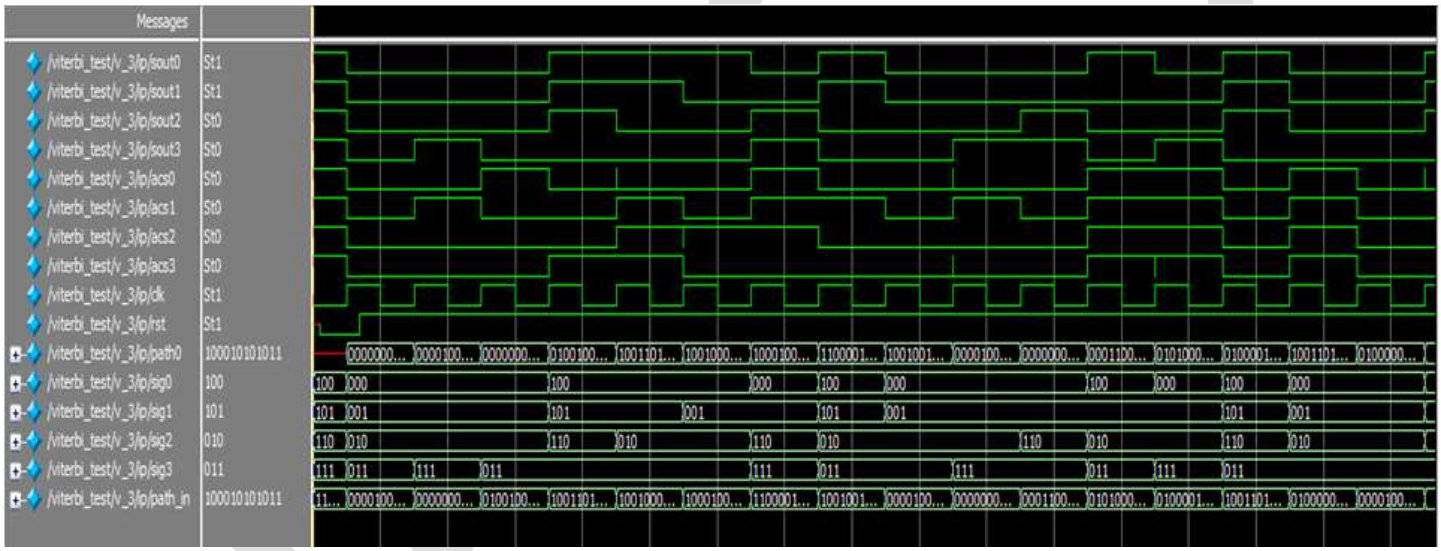

Figure11. Path in Module

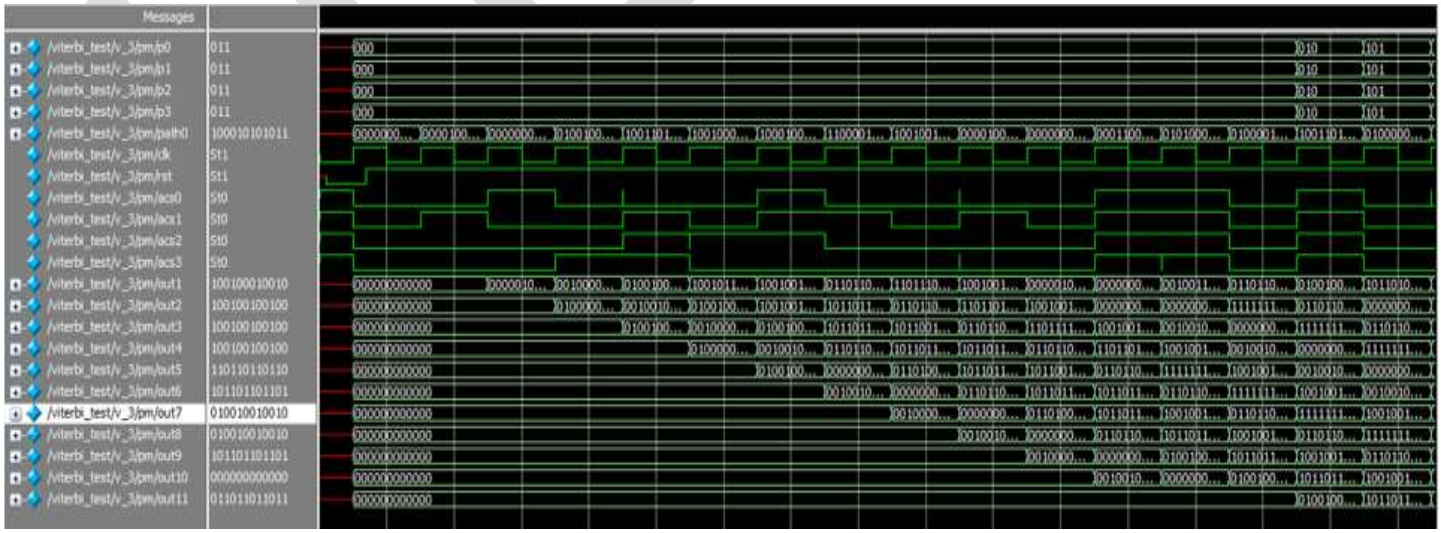

Figure12. Path Memory Module

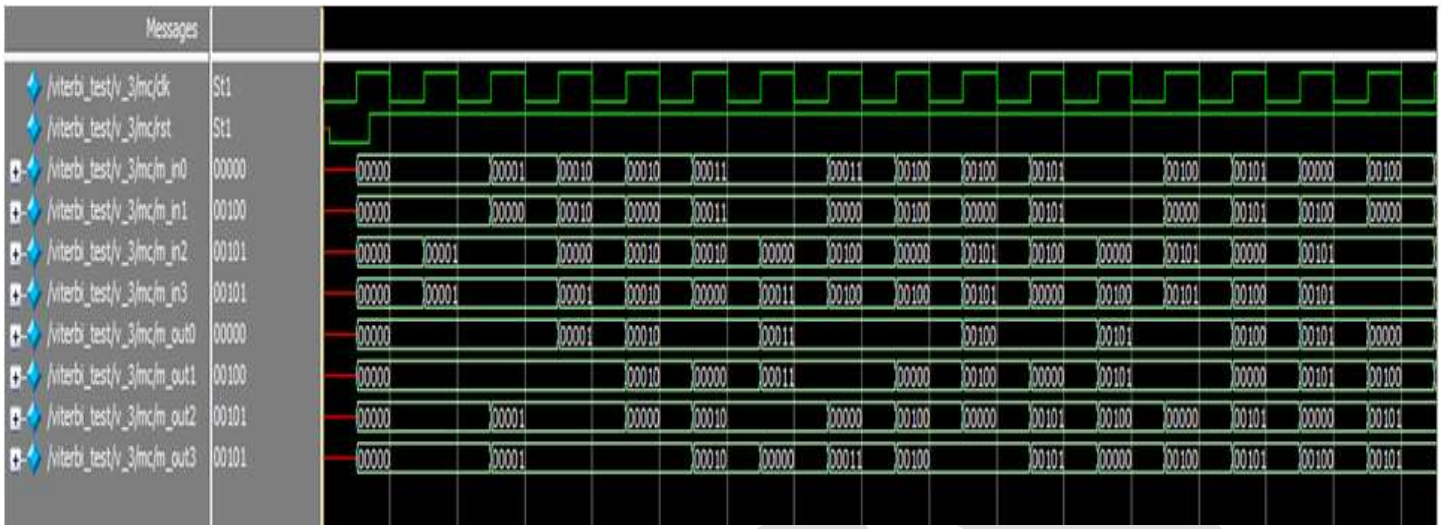

Figure13. Metric Module

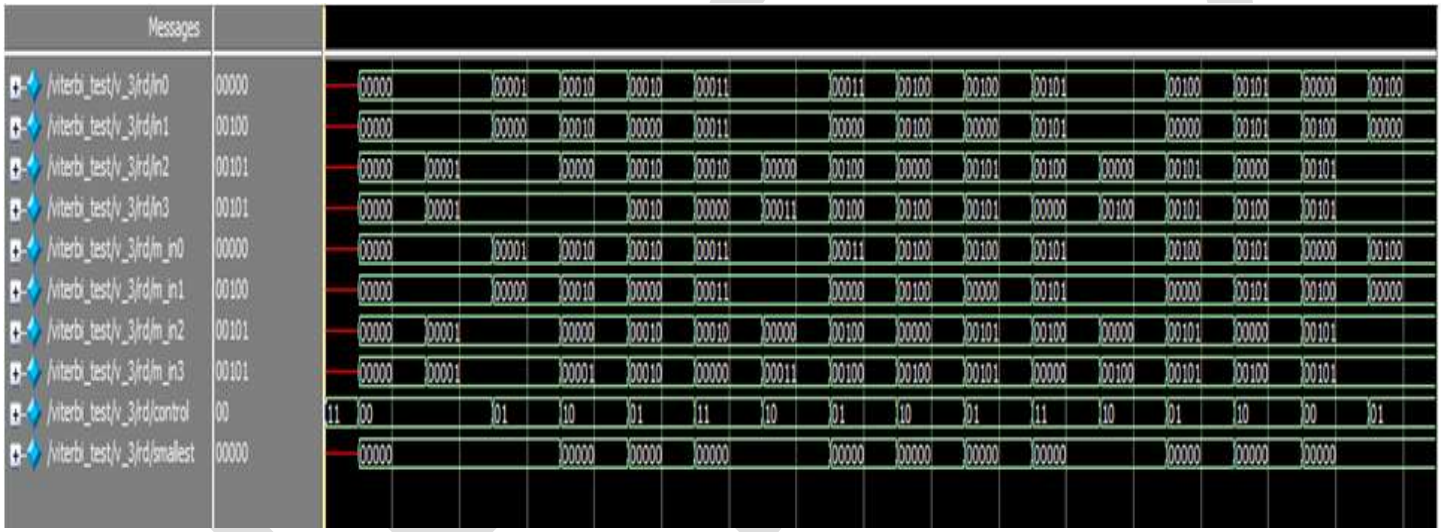

Figure14. Reduce Module

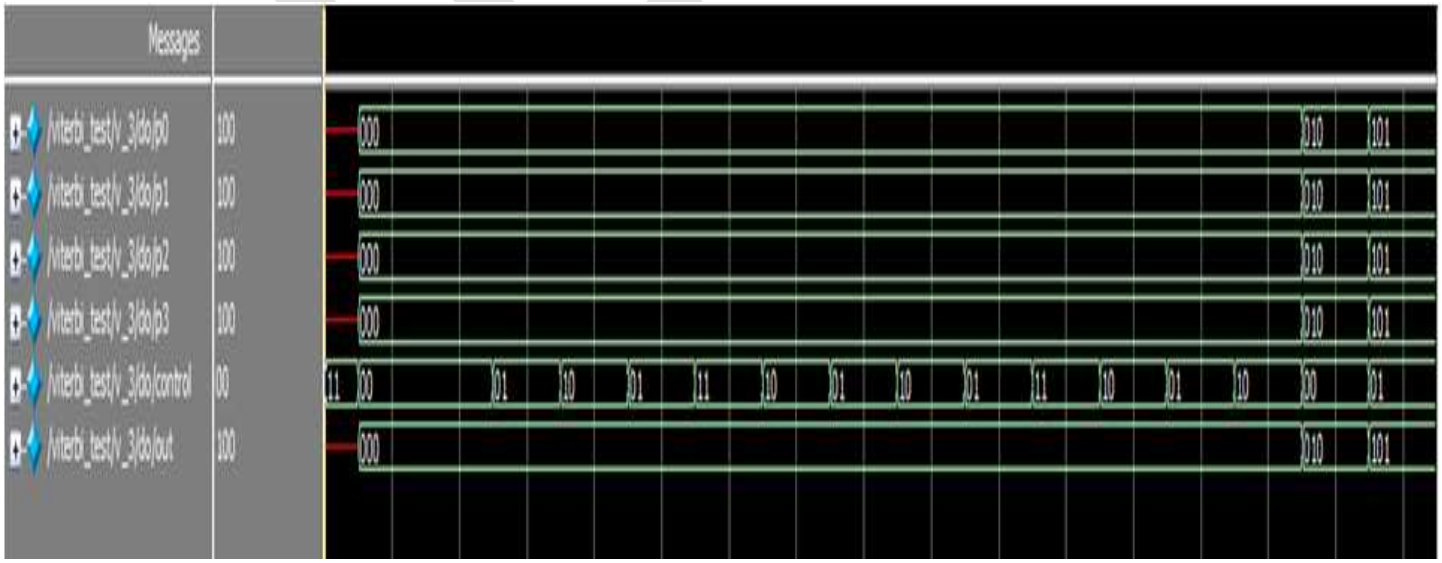

Figure15. Output Decision Module

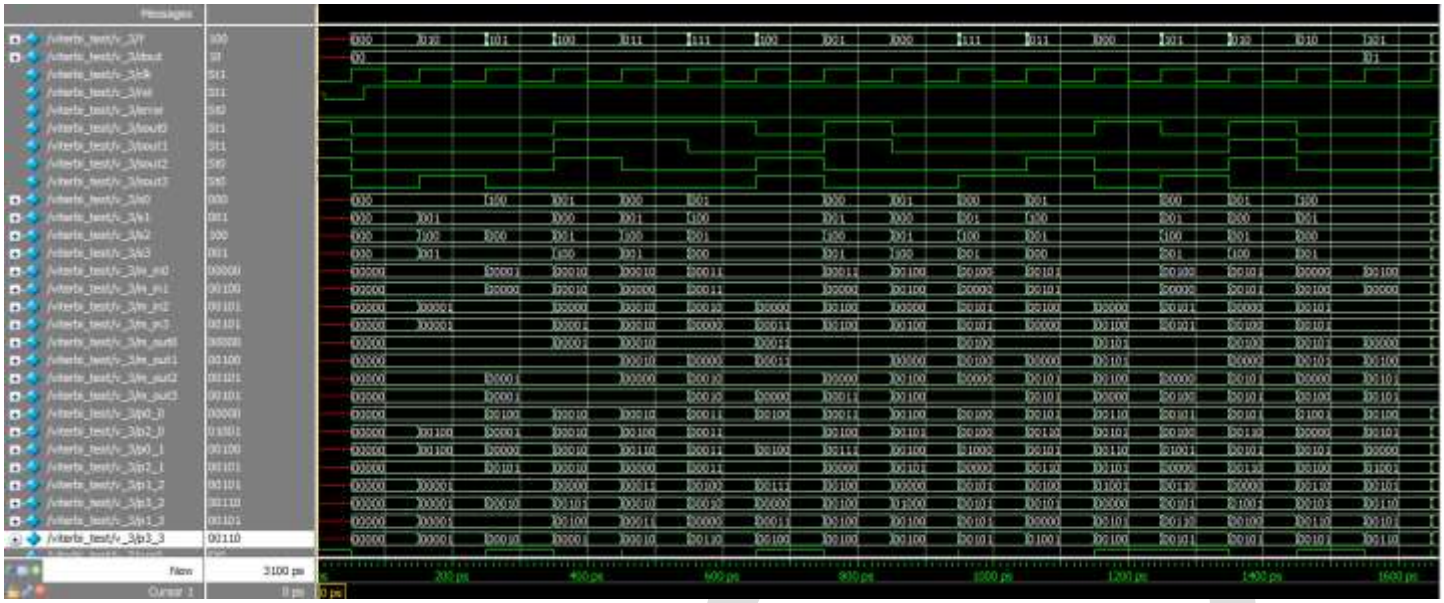

Figure16. Test bench of Viterbi decoder

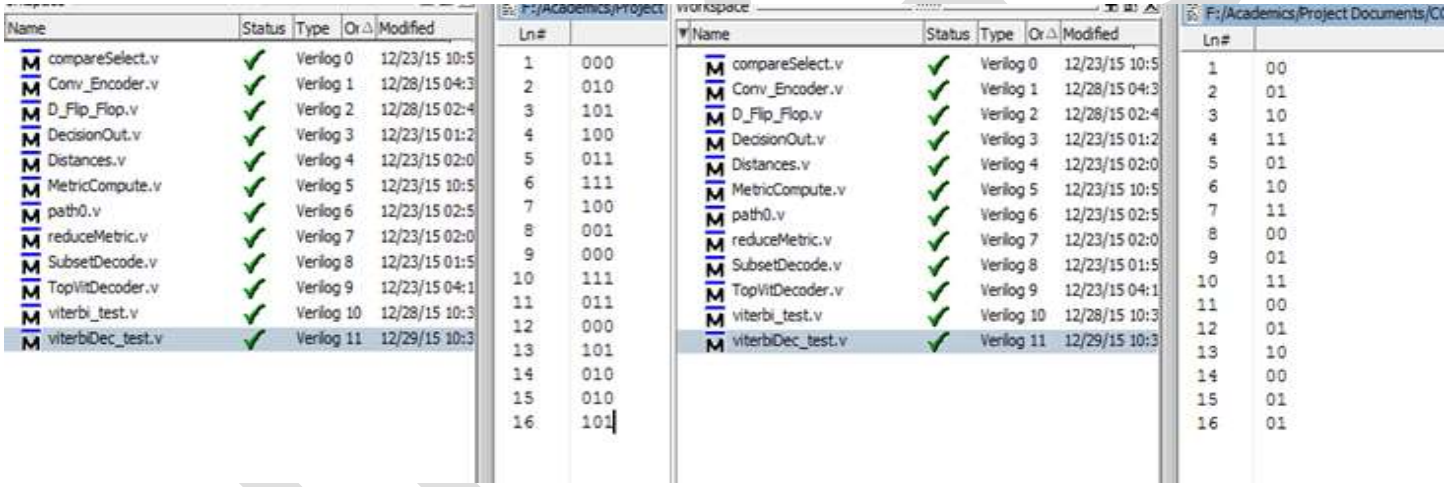

Figure17. Input and Output of Viterbi decoder

CONCLUSION

A Viterbi algorithm based on the strongly connected trellis decoding of binary convolutional codes has been presented. The use of error-correcting codes has proven to be an effective way to overcome data corruption in digital communication channels. The Viterbi decoder is modelled using Verilog, and Simulated by Xilinx ISE .We can implement a higher performance Viterbi decoder with such an algorithm. So in the future, with this algorithm with larger code rates we can get better results.

REFERENCES:

[1] Mandwale A. J and Mulani A O" Different Approaches For Implementation of Viterbi decoder on reconfigurable platform"International Conference on Pervasive Computing (ICPC) 2015

[2] P R Pachlegaonkar, S S Patki "Implementation of Reconfigurable Convolutional Encoder and optimum Adaptive Viterbi Decoder with Multi booting and Error Detection on FPGA", International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol. 3 Issue 10, October- 2014

[3] Mahe Jabeen, Salma Khan," Design of Convolution Encoder and Reconfigurable Viterbi Decoder" International Journal of Engineering and Science ISSN: 2278-4721, Vol. 1, Issue 3 (Sept 2012), PP 15-21

[4] Yan Sun, Zhizhong Ding "FPGA Design and Implementation of a Convolutional Encoder and a Viterbi Decoder Based on 802.11a for OFDM", Wireless Engineering and Technology, 2012, 3, 125-131.

[5] Karim, M.U., Khan, M.U.K. and Khawaja, Y.M. "An area reduced, speed optimized implementation of Viterbi decoder", International Conference on Computer Networks and Information Technology (ICCNIT), pp. 93-98, 2011
[6] Guan, M. and Yang, M. "Comparison and design of decoder in B3G Mobile communication system", Communications and Network, pp. 20-24, 2009
[7] Arun and V. Rajamani,"DESIGN AND VLSI IMPLEMENTATION OF A LOW PROBABILITY OF ERROR VITERBI DECODER"IEEE, First International Conference on Emerging Trends in Engineering and Technology,2008
[8] Russell Tessier, Sriram Swaminathan, Ramaswamy, Dennis Goeckel and Wayne Burleson. "A Reconfigurable, power-efficient adaptive Viterbi decoder", IEEE Transactions on Very Large Scale Integration (VLSI) systems, Vol. 13, No. 4, pp. 484-488, 2005
[9] Engling yeo, Augsburger, S.A., Rhett Davis, W. and Borivoje Nikolic . " A 500-Mb/s soft-output Viterbi decoder", IEEE Journal of solid-state circuits ,Vol. 38, No. 7, pp. 1234-1241,2003
[10] M. Kivioja, J. Isoaho and L.Vanska, "Design and implementation of Viterbi decoder with FPGAs," Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology, Kluwer Academic Publishers, vol. 21, no. 1, pp. 5-14, May 1999.
[11] C. Berrou, P. Adde, E. Angui and S. Faudeil "A Low Complexity Soft-Output Viterbi Decoder Architecture," *IEEE Int. Conf.Communications*, vol. 2, pp. 737–740, May. 1993
[12] Viterbi.A.J, "Convolution codes and their performance in communication systems," *IEEE Transaction on Communications*, vol.com-19, pp. 751 to 771, October 1971
[13] Samirkumar Ranpara, "On a Viterbi decoder design for low power dissipation," towards his master"s thesis submitted to virginia polytechnic institute and state university.
[14] B. Sklar, "Digital Communications, Fundamentals and Applications", Second Edition, New Delhi, Pearson Education, 2004
[15] Chip fleming, "A tutorial on Convolutional coding with viterbi decoding".
[16] Samir Palnitkar, "Verilog HDL: A Guide to Digital Design And Synthesis, Second Edition", Prentice Hall PTR
[17] Text book by Simon Haykin and Michael Moher : Modern Wireless Communications; Pearson Prentice Hall, 2005
[18] Text book on Applied Coding and Information Theory for Engineers by Richard B. Wells, Prentice Hall, Information and system science series