ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Novel Implementation of Modified Viterbi Decoder

Irgam.Sowjanya
VLSI System Design,
Department of ECE,
CMR Institute of Technology.

Mr.A.sai Kumar Goud
Associate Professor,
Department of ECE,
CMR Institute of Technology.

Mrs.C.Santhi
Assistant Professor,
Department of ECE,
CMR Institute of Technology.

Dr.N.V.Koteswara Rao
Professor,
Department of ECE,
CBIT, Hyderabad.

ABSTRACT:

This paper focuses on the realization of an efficient logic design of a crypto system. The type of crypto system considered in this paper is convolutional encoder and adaptive Viterbi decoder (AVD) with a constraint length, K of 3 and a code rate (k/n) of 1/3 using field programmable gate array (FPGA) technology. Here, the features of Convolutional encoder and decoder architecture are introduced and the way it can be implementable as an ASIC. Here the Viterbi Decoder is designed for faster decoding speed and less routing area with a special path management unit. The system is realized using Verilog HDL. It is simulated and synthesized using Modelsim Altera Starter Edition 6.6d and Xilinx 13.2 for RTL Design.

KEYWORDS: Convolutional Encoder, Viterbi Decoder, Verilog HDL, FPGA, PNPH Unit.

I. INTRODUCTION:

Convolutional encoding is considered to be one of the forward error correction schemes. This coding scheme is often used in the field of deep space communications and more recently in digital wireless communications. Adaptive Viterbi decoders are used to decode Convolutional codes. This technique is used for error correction. It is very efficient and robust. The main advantage of Viterbi Decoder is it has fixed decoding time and also it suites for hardware decoding implementation. But theimplementation requires the exponential increase in the area and power consumption to achieve increased decoding accuracy. Convolution coding with Viterbi decoding is a FEC technique that is particularly suited to a channel in which transmitted signal is corrupted mainly by additive white Gaussian noise (AWGN).

In most of real time applications like audio and video applications, the Convolutional codes are used for error correction. Most of the Viterbi decoders in the market are a parameterizable intelligent property (IP) core with an efficient algorithm for decoding of one convolutionally-encoded sequence only and this is mostly used in the field of satellite and radio communications.In addition, the cost for the convolutional encoder and Viterbi decoder are expensive for a specified design because of the patent issue. Therefore, to realize an adaptive Convolutional encoder and Viterbi decoder on a field programmable gate array (FPGA) board is very demanding. In this paper, we concern with designing and implementing a convolutional encoder and Viterbi decoder which are the essential block in digital communication systems using FPGA technology.Convolutional codes offer an alternative to block codes for transmission over a noisy channel. Convolutional coding can be applied to a continuous input stream (which cannot be done with block codes), as well as blocks of data. The overall block diagram of this paper is shown in the fig. 1. The message bits are generated and they are sent to the crypto system and then the decoded output is obtained.The type of crypto system used is Convolutional Encoder and Viterbi Decoder. The Convolutional encoder encodes the message and then the encoded bits are generated. The bits which are encoded are again sent to the Viterbi Decoder and then the decoded output is obtained.

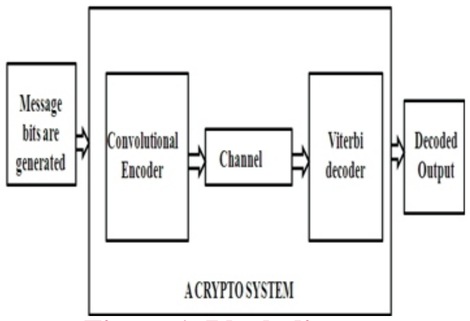


Figure.1. Block diagram

This paper deals with the implementation of the Crypto system using Verilog HDL. Section I deals with the basic block diagram of the project. Section II deals with the important definitions and terms used in the paper. Section III discusses about the Convolutional encoder and its operation. Section IV discusses about the Viterbi Decoder. Section V discusses about the software implementation. Section VI shows the Experimental results and its discussion.

II. IMPORTANT TERMS AND ITS DEFINITIONS:

The following terms are vital to the understanding of Convolutional coding and viterbi decoding.

1. Hard-decision/soft-decision decoding:

Hard-decision decoding means that the demodulator is quantized to two levels: zero and one. If you derive more than two quantization levels from the demodulator, then the decoder is soft-decision decoding.

2. Code rate R(=k/n):

Number of bits into Convolutional encoder (k)/ number of bits in output symbol which corresponds not only current input bit, but also previous (K –1) ones. It is also defined as the ratio of the number of output bits to the number of input bits.

3. Constraint length K:

It denotes the "length" of the Convolutional encoder, i.e., no of k bit stages are available to feed the combinatorial logic that produces the output symbols.

4. Branch Metric:

Difference between the received sequence and the branch word is called the Branch metric.

5. Path Metric:

Branch metric accumulates to form Path metric.

III. CONVOLUTIONAL ENCODER:

Convolutional coding has been used in communication systems including deep space communications and wireless communications. A convolutional code is used as r an alternative to block codes of transmission.

Convolutional coding can be applied to a continuous input stream (which cannot be done with block codes), as well as blocks of data. A convolutional encoder is a Mealy machine, where the output is a function of the current state and the current input. It consists of one or more shift registers (DFF) and multiple XOR gates. XOR gates are connected to some stage of the shift registers as well as to the current input to generate the output polynomials.

The encoder in figure 2 produces two bits of encoded information for single bit of input information, so it is called a rate 1/2 encoder. A Convolutional encoder is generally represented .

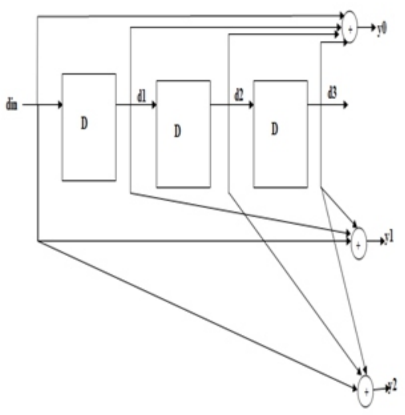


Figure.2. Convolutional encoder of rate 1/3

in (n, k, m) format with a rate of k/n, where n is number of outputs of the encoder, k is number of inputs of the encoder, m is number of flip-flops. In this paper, we discuss decoding of convolutional codes generated by a (n,1,m) encoder with the rate 1/n with the number of outputs n=3. A Convolutional encoder is a Mealy machine, where the output is a function of the current state and the current input. It consists of one or more shift registers and multiple XOR gates.The above figure 2 shows the Convolutional Encoder with rate =1/3. Convolutional encoder can be described in terms of state table, state diagram and trellis diagram.

IV. VITERBI DECODER:

This section discusses the different parts of the Viterbi decoder. It performs basically two operations, they are Synchronization and Quantization. Analog signals are quantized and converted into digital signals in the quantization block. The synchronization block detects the frame boundaries of code words and symbol boundaries.

The Viterbi decoder receives successive code symbols, in which the boundaries of the symbols and the frames have been identified. The viterbi decoder is basically divided into two types which are discussed below.

A. Hard decision viterbi decoding:

In the hard-decision decoding, the path through the trellis is determined using the Hamming distance measure. Thus, the most optimal path through the trellis is the path with the minimum Hamming distance. In this type of decoding it shows only two possibilities, one is the presence of signal and the other is absence of signal i.e either '1' or '0'.The Hamming distance can be defined as a number of bits that are different between the observed symbol at the decoder and the sent symbol from the encoder. Furthermore, the hard decision decoding applies one bit quantization on the received bits.

B. Soft decision viterbi decoding:

Soft-decision decoding is applied for the maximum likelihood decoding, when the data is transmitted over the Gaussian channel. On the contrary to the hard decision decoding, the soft decision decoding uses multi-bit quantization for the received bits, and Euclidean distance as a distance measure instead of the hamming distance.

C. Internal Architecture of Viterbi Decoder:

A hardware Viterbi decoder for basic (not punctured) code usually consists of the following major blocks:

- Branch metric unit (BMU)
- Path metric unit (PMU)
- Trace back unit (TBU)

The block diagram of the hardware viterbi decoder is shown in fig. 3.

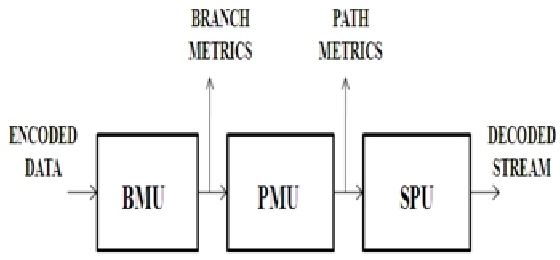


Figure.3. Hardware Viterbi Decoder

In the above diagram the encoded input is given to the branch metric unit which gives the branch metrics, and these branch metrics are again given to the path metric unit which again produces the path metrics from which using trace back method the decoded stream is obtained.

i. Branch metric unit:

The first unit is called branch metric unit. Here the received data symbols are compared to the ideal outputs of the encoder from the transmitter and branch metric is calculated. Hamming distance or the Euclidean distance is used for branch metric computation. The sample implementation of branch metric calculation unit is shown in in the fig. 4.

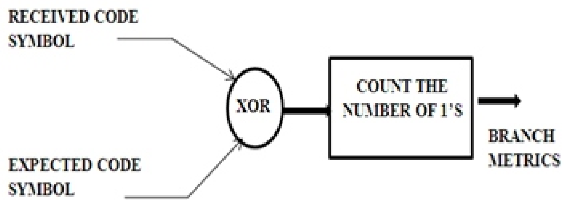


Figure.4. Branch metric calculation

A branch metric unit's function is to calculate branch metrics, which are normed distances between every possible symbol in the code alphabet, and the received symbol. The branch metrics are difference values between received code symbol and the corresponding branch words from the encoder trellis.

ii. Path metric unit:

The PMU calculates new path metric values and decision values. Because each state can be achieved from two states from the earlier stage, there are two possible path metrics coming to the current state. The core elements of a PMU are ACS (Add-Compare-Select) units. The way in which they are connected between themselves is defined by a specific code's trellis diagram. The ACS unit, as shown in fig. 5, adds for each of the two incoming branches the corresponding states path metric, resulting in two new path metrics. The path with the better metric is chosen and stored as the new path metric for current state, while generating a decision bit.

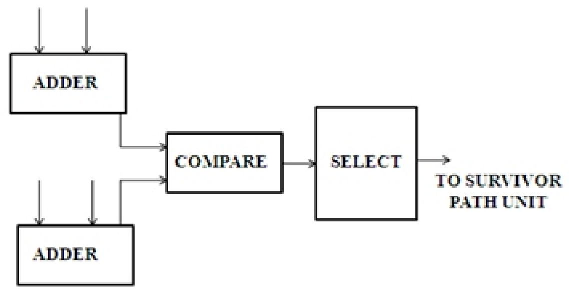


Figure.5. Add compare Select unit (ACSU)

A path metric unit summarizes branch metrics to get metrics for 2K-1 paths, where K is the constraint length of the code, one of which can eventually be chosen as optimal. Every clock it makes 2K-1 decisions, throwing off wittingly non optimal paths. The results of these decisions are written to the memory of a trace back unit.

iii. Survivor Path unit:

The survivor path unit stores the decisions of the ACS unit and uses them to compute the decoded output. The trace-back technique and the register-exchange approaches are two major techniques used for the path history management. The Trace back unit takes up less area but require much more time than the Register Exchange method.

A relatively new approach called permutation network based path history unit implements directly the trellis diagram of the given Convolutional code to trace the survivor path back sequentially.The resulting circuit has smaller routing area than register-exchange technique and has faster decoding speed than trace-back method f regardless of the constraint length.

iv. Permutation network based path history (PNPH) unit:

The PNPH (Permutation Network based Path History) unit for an convolutional code is a 5L-stage permutation network with each stage containing 1-to-2k demultiplexers, where each DeMux corresponds to each node of the trellis diagram and is associated with a K-bit register and a 2k-input OR gate[2]. The K bit register is used to store the decision bits associated with the state node and to determine the partial survivor path associated with the node[2].

Thus, each registers demultiplexer pair determines the part of the survivor path associated its corresponding state node [2]. The connection between two adjacent stages of the interconnection network is defined by the next function of the state diagram of the underlying encoder.

New decision-bit values for each state calculated by add-compare-select (ACS) enter into the rightmost end of corresponding shift register. The PNPH unit for the convolutional code (3, 1, 3) shown in Fig. 6 below [2].

V. SIMULATION OF VITERBI ALGORITHM:

The various modules of Convolutional encoder and Viterbi decoder in verilog HDL is implemented here. The whole process of convolutional encoder and Adaptive Viterbi decoder can be summarized as shown in the flowchart shown in figure 7.

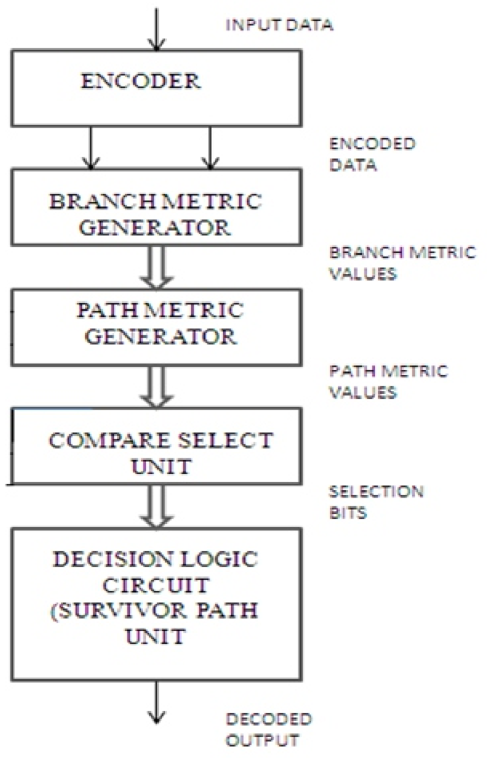


Figure.6. Flowchart of the convolutional encoder and viterbi decoder implementation.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

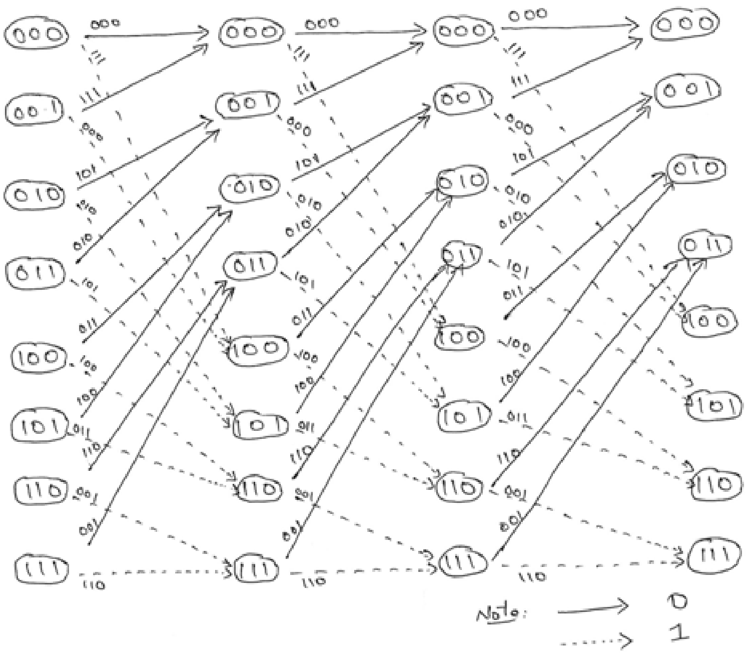
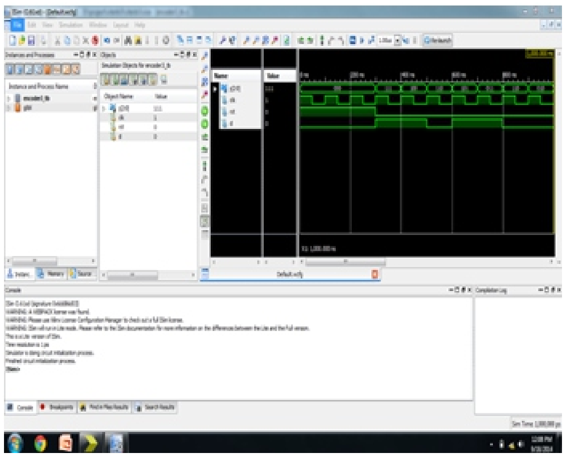


Figure 7: Trellis Coded Modulation

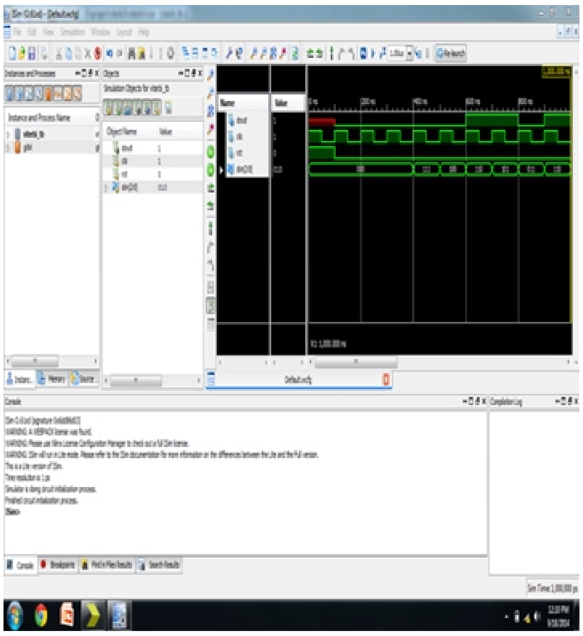
VI EXPERIMENTAL RESULTS

Simulation Results:



```
ISim O.61xd (signature [illegible])
WARNING: A WEBPACK license was found.
WARNING: Please use Xilinx License Configuration Manager to check out a full ISim license.
WARNING: ISim will run in Lite mode. Please refer to the ISim documentation for more information on the differences between the Lite and the Full version.
This is a Lite version of ISim.
Time resolution is 1 ps
Simulator is doing circuit initialization process.
Finished circuit initialization process.
ISim>
```

Figure 8: Simulation Results of 3 bit Adaptive Viterbi encoder



```
ISim O.61xd (signature [illegible])
WARNING: A WEBPACK license was found.
WARNING: Please use Xilinx License Configuration Manager to check out a full ISim license.
WARNING: ISim will run in Lite mode. Please refer to the ISim documentation for more information on the differences between the Lite and the Full version.
This is a Lite version of ISim.
Time resolution is 1 ps
Simulator is doing circuit initialization process.
Finished circuit initialization process.
ISim>
```

Figure 9: Simulation Results of 3 bit Adaptive Viterbi decoder.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

RTL Schematic:

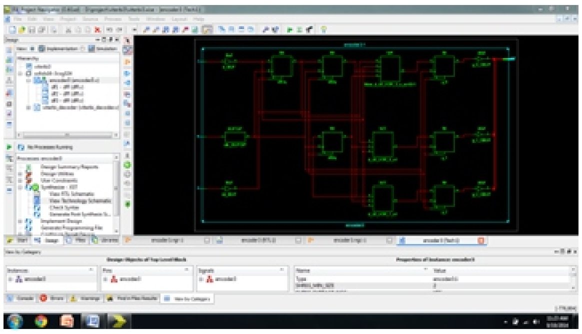


Figure 10: RTL Schematic of 3 bit Adaptive Viterbi Encoder.

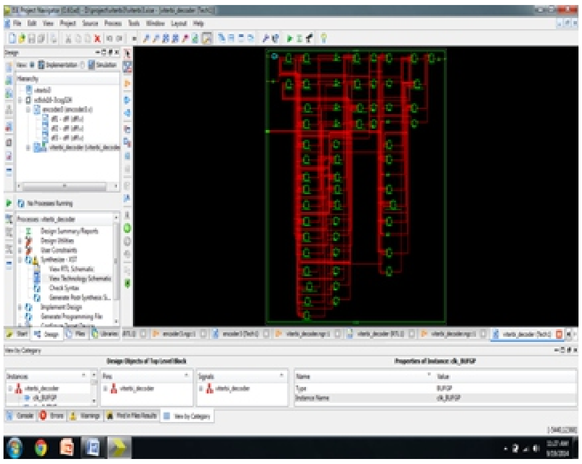
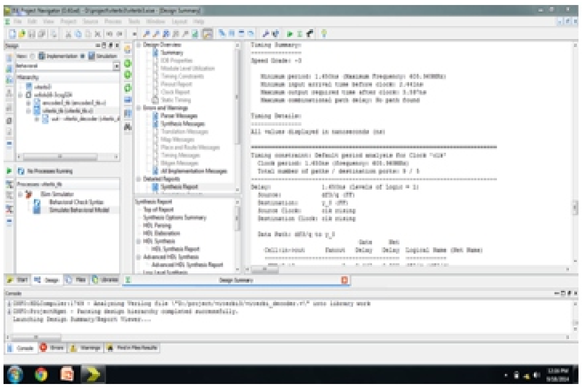


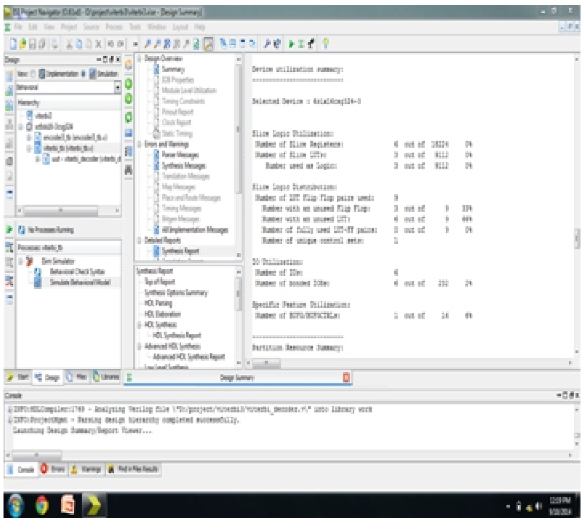
Figure 11: RTL Schematic of 3 bit Adaptive Viterbi Decoder

Timing Report



```
Timing Summary:
---------------
Speed Grade: -3

   Minimum period: 1.450ns (Maximum Frequency: [illegible]MHz)
   Minimum input arrival time before clock: 2.441ns
   Maximum output required time after clock: 3.597ns
   Maximum combinational path delay: No path found

Timing Details:
---------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default period analysis for Clock 'clk'
  Clock period: 1.450ns (frequency: [illegible]MHz)
  Total number of paths / destination ports: 3 / 3
-------------------------------------------------------------------------
Delay:               1.450ns (Levels of Logic = 1)
  Source:            [illegible] (FF)
  Destination:       y_3 (FF)
  Source Clock:      clk rising
  Destination Clock: clk rising

  Data Path: [illegible] to y_3
                                Gate     Net
    Cell:in->out      Fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
```

Device Utilization Report:



```
Device utilization summary:
---------------------------

Selected Device : 6slx16csg324-3


Slice Logic Utilization:
 Number of Slice Registers:              6  out of  18224     0%
 Number of Slice LUTs:                   3  out of   9112     0%
    Number used as Logic:                3  out of   9112     0%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:     9
   Number with an unused Flip Flop:      3  out of      9    33%
   Number with an unused LUT:            6  out of      9    66%
   Number of fully used LUT-FF pairs:    0  out of      9     0%
   Number of unique control sets:        1

IO Utilization:
 Number of IOs:                          6
 Number of bonded IOBs:                  6  out of    232     2%

Specific Feature Utilization:
 Number of BUFG/BUFGCTRLs:               1  out of     16     6%

---------------------------
Partition Resource Summary:
```

```
INFO:HDLCompiler:1749 - Analyzing Verilog file \"D:/project/viterbi3/viterbi_decoder.v\" into library work
INFO:ProjectMgmt - Parsing design hierarchy completed successfully.
Launching Design Summary/Report Viewer...
```

Figure 11: Device Utilization Report of 3 bit Adaptive Viterbi Decoder.

VII. CONCLUSION:

In this paper, we have presented the design and implementation of the Convolutional encoder and Viterbi decoder. This design has been simulated in MODELSIM altera 6.6d and synthesized using XILINX-ISE 12.4i for the constraint length of K=7 and code rate of ½ input sequence. The given input sequence has been encoded by using convolutional encoder and it is transmitted through the channel. Finally, the transmitted sequence is decoded by the Viterbi decoder and the estimated original sequence is produced.

REFERENCES:

[1] Anh Dinh, Ralph mason and Joe toth," High speed v.32 Trellis encoder & decoder implementation using FPGA,"in IEEE transactions on communications,1999.

[2] Ming-Bo Lin, "New path history management circuits for viterbi decoders," in IEEE transactions on communications, vol 48, no.10, october 2000.

[3] A.J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," ieee transactions on information theory, vol. it-13, april, 1967, pp. 260-269.

[4] Samirkumar Ranpara,"On a Viterbi decoder design for low power dissipation," towards his master's thesis submitted to virginia polytechnic institute and state university.

[5] Chip fleming,"A tutorial on Convolutional coding with viterbi decoding" .

[6] S.Lin and D.J. Costello, Error control coding. Englewood cliffs, nj: prentice hall, 1982.

[7] Wong, Y.S. et.al "Implementation of convolutional encoder and Viterbi decoder using VHDL" IEEE Tran. on Inform. Theory, Pp. 22-25, Nov. 2009.

[8] Hema.S, Suresh Babu.V and Ramesh.P, "FPGA Implementation of Viterbi decoder" proceedings of the 6th WSEAS ICEHWOC,Feb. 2007.

[9] Irfan Habib, Özgün Paker, Sergei Sawitzki, "Design Space Exploration of Hard-Decision Viterbi Decoding: Algorithm and VLSI Implementation" IEEE Tran. On Very Large Scale Integration (VLSI) Systems, Vol. 18, Pp. 794-807, May 2010.