

FPGA Implementation of Convolutional Encoder and Adaptive Viterbi Decoder

B. SWETHA REDDY¹, K. SRINIVAS²

¹PG Scholar, Dept of ECE, Aurora's Technological and Research Institute, JNTU Hyderabad, Telangana, India,
E-mail:swetha.basireddy.15@gmail.com.

²Assistant Professor, Dept of ECE, Aurora's Technological and Research Institute, JNTU Hyderabad, Telangana, India,
E-mail: kotha.srinivas.428@gmail.com.

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 7 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 Xilinx ISE 13.2 for RTL Design.

Keywords: Convolutional Encoder, Adaptive Viterbi Decoder (AVD), VHDL, Viterbi Algorithm, FPGA.

I. INTRODUCTION

Most digital communication systems nowadays convolutionally encoded the transmitted data to compensate for Additive White Gaussian Noise (AWGN), fading of the channel, quantization distortions and other data degradation effects. For its efficiency the Viterbi algorithm has proven to be a very practical algorithm for forward error correction of convolutionally encoded messages [1]. The requirements for the Viterbi decoder or Viterbi detector depend on the applications used. Most of the researches work to reduce cost, the power consumption, or work with high frequency for using the decoder in the modern applications. The following discovery code was the cyclic codes. Cyclic codes are also called cyclic redundancy check (CRC) codes primarily used today for the error detection applications rather than for error correction [5]. Bose- Chaudhuri-Hocquenghem (BCH) codes are the important subclass of the cyclic codes which discovered by Hocquenghem in 1959 and by the team of Bose and Ray-Chaudhuri in 1960. Then the BCH codes were extended to the non-binay case ($q > 2$) by Reed and Solomon in 1960. All communication channels are subject to the additive white gaussian noise (AWGN) around the environment. Forward error correction (FEC) techniques are used in the transmitter to encode the data stream and receiver to detect and correct bits in errors, hence minimize the bit error rate (BER) to improve the performance.

RS decoding algorithm complexity is relatively low and can be implemented in hardware at very high data rates. It seems to be an ideal code attributes for any application. However, RS codes perform very poorly in AWGN channel. Due to weaknesses of using the block codes for error correction in useful channels, another approach of coding

called convolutional coding had been introduced in 1955 [7]. convolutional encoding with Viterbi decoding is a powerful FEC technique that is particularly suited to a channel in which the transmitted signal is corrupted mainly by AWGN [6]. It operates on data stream and has memory that uses previous bits to encode. It is simple and has good performance with low implementation cost. The Viterby algorithm (VA) was proposed in 1967 by Andrew Viterbi [8] and is used for decoding a bitstream that has been encoded using FEC code. The Convolutional encoder adds redundancy to a continuous stream of input data by using a linear shift register.

II. LITERATURE SURVEY

Verilog HDL programming language is used for project. Verilog HDL project is portable. Being created for one element base, a computing device project can be ported on another element base, for example VLSI with various technologies. Verilog HDL is commonly used to write text models that describe a logic circuit. Such a model is processed by a synthesis program, only if it is part of the logic design. A simulation program is used to test the logic design using simulation models to represent the logic circuits that interface to the design. This collection of simulation models is commonly called a test bench. Verilog HDL is a hardware description language that can be used to model a digital system at many levels of abstraction, ranging from the algorithmic level to the gate level. An in depth study is also done. The Verilog HDL language can be regarded as an integrated amalgamation of the following languages:

- Sequential language
- Concurrent language
- Net-list language

- Timing specifications

This language not only defines the syntax but also defines very clear simulation semantics for each language construct. Therefore, models written in this language can be verified using a Verilog HDL simulator. This subset is usually sufficient to model most applications. The complete language, however, has sufficient power to capture the descriptions of the most complex chips to a complete electronic system. One can design hardware in a Verilog HDL IDE (for FPGA implementation such as Xilinx ISE, Altera Quartus, Synopsys Synplify or Mentor Graphics HDL Designer) to produce the RTL schematic of the desired circuit. After that, the generated schematic can be verified using simulation software which shows the waveforms of inputs and outputs of the circuit after generating the appropriate test bench. To generate an appropriate test bench for a particular circuit or Verilog HDL code, the inputs have to be defined correctly. For example, for clock input, a loop process or an iterative statement is required.

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) [3]. In most of real time applications like audio and video applications, the convolutional codes are used for error correction [2]. Convolutional code definition parameters are the following: code rate (R), generating polynomial $g(n)$, and number of input bits (k), number of output bits (n) and constraint length (K). The code rate is the number of transmitted bits per input bit, e.g., a rate 1/2 encodes 1 bit and produces 2 bits for transmission. One generating polynomial stands for one output. The constraint length is the length of the generating polynomial in bits. The higher it is the more robust is the code. Passing the information sequence to be transmitted through a linear finite shift register generates a convolutional code. The shift register consists of k bit stages and n linear algebraic function generators. The content of shift register is multiplied by respective term in generator matrix and is then added together to generate respective code words.

The information symbols are encoded by using a convolution operation. Symbols, which are defined by coefficients in the generator polynomial, are added modulo 2 to each other and form output signal [3]. 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 [3]. The basic building blocks of Viterbi decoder are branch metric unit (BMU), path metric unit (PMU), add compare and select unit (ACSU) and state machine control unit (SMC). However, the complexity of the Viterbi decoder increases exponentially with the constraint length, so it is impractical to use codes with constraint lengths more than $K > 15$ (3 to 9 is common practice) [1]. Viterbi decoders employed in digital wireless communications are complex and dissipate large power. Sometimes, due to incomplete design space exploration or incorrect analysis, a suboptimal design is chosen. This work analyzes the design complexity by applying most of the known VLSI implementation techniques for hard-decision Viterbi decoding to a different set of code parameters.

Recently, convolutional codes have become more and more important in digital transmission. A convolutional code with Viterbi decoding is used in wireless communication and satellite communication. The various examples are cellular phone i.e. GSM, IS-54 digital cellular phone standards and IS-95 CDMA standard, modems and video and audio broadcasting. Motivation for efficient hard decision decoder has been derived from needs to increase the speed, to extend the battery life and to reduce the cost. Viterbi decoder is one of the most widely used components in digital communications and storage devices. Although its software implementation is studied in depth over the last decades (Matlab, embedded C) still every new design starts with design space exploration. This can be partially explained by the fact that the design space is huge. In addition, the optimization criteria and the design figures keep on changing with the advancement in semiconductor processing technology and design tools. Different design aspects of the Viterbi decoder have been studied in a number of research papers. However, most researchers concentrate on one specific component of the design (e.g., path metrics unit or survival memory unit). Somewhat more general studies are presented in [1].

The proposed research work will carry out on re-configurable FPGA technology, by adopting parallel/pipeline features of the hardware resources. The overall system performance could be improved. The existing algorithm is redesigned using HDL language, simulation, synthesis and implementation (translation, mapping place & routing) done with FPGA [12-14] based EDA tools. The Convolutional Encoder and Viterbi Decoder used in the Digital Communications System is shown in Fig. 1. By exploring the design at architecture, RTL level and selecting desired optimization constraints (area, speed, power) to Selected FPGA device synthesis technology, the system performance could be improved and comparison analysis could be carried out with old existing research works. This implements the

FPGA Implementation of Convolutional Encoder and Adaptive Viterbi Decoder

error correction method at real time while operating many-to-one and one-to-many data communication. The overall architecture primitive modules are designed, simulated, synthesized and structured in VHDL and by using various FPGA based EDA Tools.

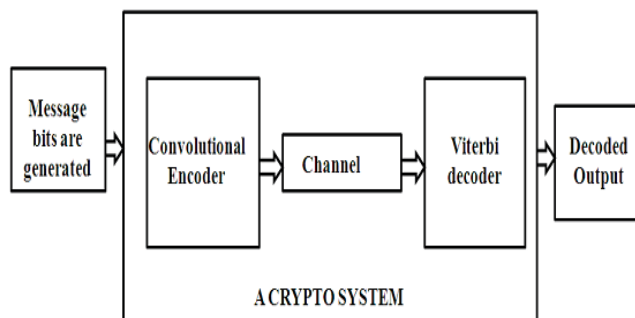


Fig.1. Digital Communication system Block Diagram.

III. CONVOLUTIONAL ENCODER

A convolutional code is a type of error-correcting code which differs a lot from block codes. First, the former does not have code words made up of distinct data sections and block sections. Instead, redundant bits are distributed throughout the coded data. Second, the encoder of the former contains memory and the n encoder outputs at any given time unit depend not only on the k inputs at that time unit but also on m previous input blocks. Convolutional codes are sometimes referred as trellis codes. Normally, convolutional encoding is simple, but decoding is much more difficult. Convolutional codes are usually characterized by two parameters and the patterns of n modulo-3 adders. The two important parameters are the code rate and constraint length. The code rate (k/n) where the number of output bits must equal or bigger than the input bits (n_k), 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. To convolutionally encode data, start with m memory registers, each holding 1 input bit. Unless otherwise specified, all memory registers start with a value of 0.

The encoder has n modulo-3 adders, and n generator polynomials, one for each adder. An input bit m_1 is fed into the leftmost register. Using the generator polynomials and the existing values in the remaining registers, the encoder outputs n bits [1]. As shown in Fig.2, where we have a general encoder designed with a code rate (k/n) of $1/3$ and an information sequence that is being shifted in to the register m of 1 bit at a time. The shift register has a constraint length (K) of 7, equal to the number of stages in the register. The output from the encoder is called code symbols. At initialization all stages in the encoder shall be initially set to zero. The output of the encoder is determined by the generator polynomial equations. Since the complexity of the encoder increases exponentially with the constraint length, none of the encoders uses more than a constraint length of 9, for practical reasons. [9] - [11].

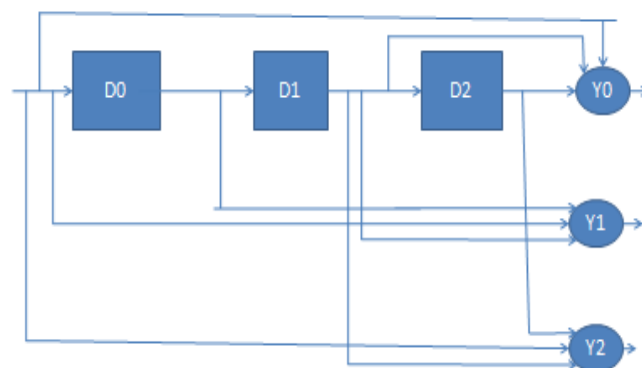


Fig.2. Convolutional Encoder.

Inputs for the encoder are D_{in} , D_1 , D_2 & D_3 .

Outputs are Y_0 , Y_1 & Y_2 .

$Y_0 = D_{in} \oplus D_1 \oplus D_2 \oplus D_3$;

$Y_1 = D_{in} \oplus D_1 \oplus D_3$;

$Y_2 = D_{in} \oplus D_2 \oplus D_3$;

IV. VITERBI ALGORITHM

The Viterbi decoding algorithm proposed in 1967 is a decoding process for Convolutional codes. Convolutional coding, as we all known, has been widely used in communication systems including deep space communications and wireless communications, such as IEEE 802.11a/g, WiMax, DAB/DVB, WCDMA and GSM. Viterbi decoding is the best technique for decoding the convolutional codes but it is limited to smaller constraint lengths. The basic building blocks of Viterbi decoder are branch metric unit, add compare and select unit and survivor memory management unit. The two techniques for decoding the data are trace back (TB) method and Register Exchange (RE) method. TB method is used for longer constraint lengths but it has larger decoding time. Also extra circuitry is required to reverse the decoded bits. The RE method is simpler and faster than the TB method for implementing the VD. RE method is not appropriate for decoders with long constraint lengths. Viterbi algorithm was introduced in 1967 by Viterbi. Viterbi algorithm is called as optimum algorithm because it minimizes the probability of error.

The algorithm can be broken down into the following three steps.

- Weigh the trellis; that is, calculate the branch metrics.
- Recursively computes the shortest paths to time n , in terms of the shortest paths to time $n-1$. In this step, decisions are used to recursively update the survivor path of the signal. This is known as add-compare-select (ACS) recursion.
- Recursively finds the shortest path leading to each trellis state using the decisions from Step 2. The shortest path is called the survivor path for that state and the process is referred to as survivor path decode. Finally, if all survivor paths are traced back in time, they merge into a unique path, which is the most likely signal path [2].

A data sequence x is encoded to generate a convolutional code word y . After y is transmitted through a noisy channel. The convolutional decoder takes the received vector r and generates an estimate z of the transmitted code word. The maximum likelihood (ML) decoder selects the estimate that maximizes the probability $p(r|z)$, while the maximum a posteriori probability (MAP) decoder selects the estimate that maximizes $p(z|r)$. If the distribution of the source bits x is uniform, the two decoders are identical. By Bayes's law, we could get:

$$p(r|z)p(z) = p(z|r) \quad (1)$$

The initial part first initializes the Viterbi decoder to the same FSM and the trellis diagram as the convolutional encoder as. Then in each time clock, the decoder computes the four possible branches metric's Euclidean distance. For each state, ACS block compute the two possible paths Euclidean distance and select a small one. At the same time, ACS block will record the survival state metric. A trellis structure is built along the way, where each state transition is noted with all the relevant information like path metric and the decision bit. When all the data bits have been received, and the trellis has been completed, the last stage of the trellis is used as the starting point for tracing back. The trace-back operation outputs the decoded data, as it traces back along the maximum likelihood path. The Viterbi algorithm operates on a state machine assumption. That is, at any time the system being modeled is in one of a finite number of states. While multiple sequences of states (paths) can lead to a given state, at least one of them is a most likely path to that state, called the "survivor path".

This is a fundamental assumption of the algorithm because the algorithm will examine all possible paths leading to a state and only keep the one most likely. This way the algorithm does not have to keep track of all possible paths, only one per state. A second key assumption is that a transition from a previous state to a new state is marked by an incremental metric, usually a number. This transition is computed from the event. The third key assumption is that the events are cumulative over a path in some sense, usually additive. So the crux of the algorithm is to keep a number for each state. When an event occurs, the algorithm examines moving forward to a new set of states by combining the metric of a possible previous state with the incremental metric of the transition due to the event and chooses the best. The incremental metric associated with an event depends on the transition possibility from the old state to the new state.

V. VITERBI DECODER ARCHITECTURE

A Viterbi decoder uses the VA for decoding a bit stream that has been encoded using FEC based on a convolutional code. The Viterbi Decoder is used in many FEC applications and in systems where data are transmitted and subject to errors before reception. The VA is commonly used in a wide range of communications and data storage applications. It is used for decoding convolutional codes, in base band detection for wireless systems, and also for detection of recorded data in magnetic disk drives. The requirements for

the Viterbi decoder or Viterbi detector, which is a processor that implements the VA, depend on the applications where they are used. The block diagram of Viterbi decoder is shown in Fig.3.

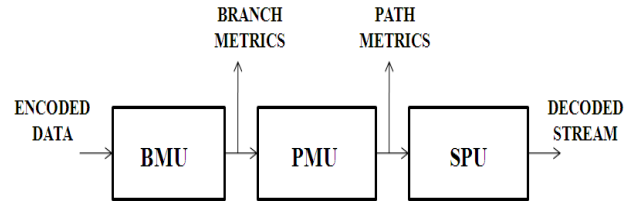


Fig.3. Viterbi decoder block diagram.

The block diagram consists of the following modules: Branch Metrics, Add-Compare-Select (ACS), register exchange, maximum path metric selection, and output register selection [9], [11].

A. The Branch Metric Calculation (BMC)

This is typically based on a look-up table containing the various bit metrics. The computer looks up the n -bit metrics associated with each branch and sums them to obtain the branch metric. The result is passed along to the path metric Calculation. The responsibility of this unit is to compute the Hamming code between the expected code and the receiving code as a frame. At each processing, the BMU finds the Hamming code for these symbols.

B. Path Metric Calculation

There are Path Metric Unit (PMU) and Add Compare Select Unit (ACSU) blocks in it.

Path Metric Unit (PMU): It computes the partial path metrics at each node in the trellis.

Add Compare Select Unit (ACSU): This ACSU is the main unit of the survivor path decoder. The function of this unit is to find the addition of the Hamming code received from BMU's and to compare the total Hamming code. This takes the branch metrics computed by the BMC and computes the partial path metrics at each node in the trellis. The surviving path at each node is identified, and the information-sequence updating and storage unit notified accordingly. Since the entire trellis is multiple images of the same simple element, a single circuit called Add- Compare-Select may be assigned to each trellis state.

C. Survivor Management Unit (SMU)

This is responsible for keeping track of the information bits associated with the surviving paths designated by the path metric Calculation. There are two basic design approaches: Register Exchange and Trace Back. In both techniques, a shift register is associated with every trellis node throughout the decoding operation. Since one of the major interests is the low power design, the proposed decoder has been implemented using the trace back approach which dissipates less power. The major disadvantage of the RE approach is that its routing cost is very high especially in the case of long-constraint lengths and it requires much more resources.

D. Trellis Diagram

A trellis diagram is a time-indexed version of a state machine, and the simplest 3-state trellis is shown in fig.4. Each state corresponds to a possible pattern of recently received data bits and each branch corresponds to a receipt of the next (noisy) input. The goal is to find the path through the trellis of maximum likelihood, because that path corresponds to the most likely pattern that the transmitter actually sent.

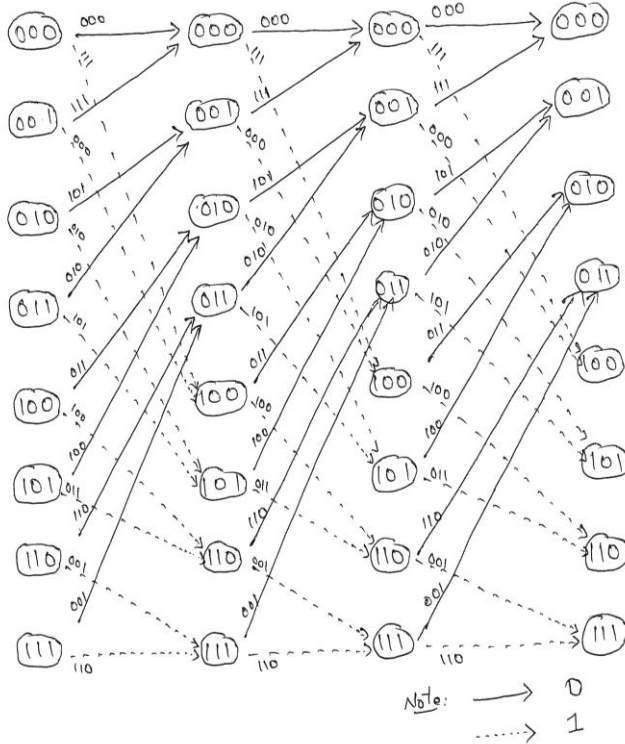


Fig.4. Three state trellis.

VI. RESULTS

A. Simulation Result

The Figs.5, 6 and 7 are the simulation result of the convolutional encoder. The first signal represents the input data given to the encoder. The next signal is the reset signal and next one is the clock signal that enables the process of the block. The signals after the clock signal represent the output with different combinations of input.

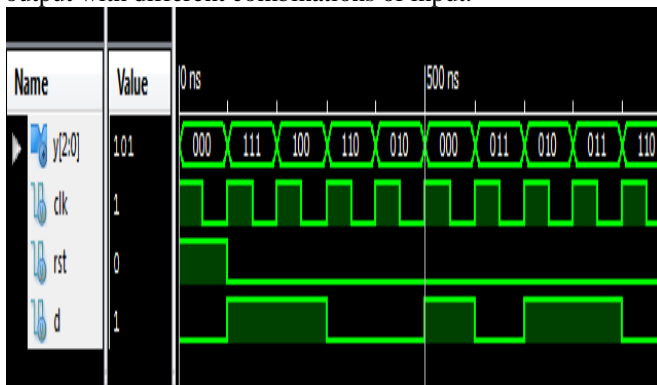


Fig.5. Simulation wave form Encoder.

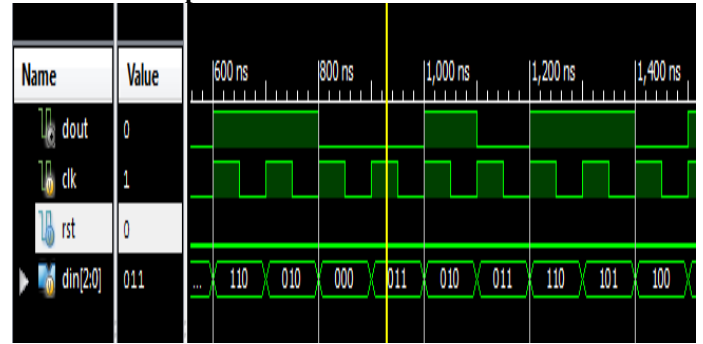


Fig.6. Simulation wave form of viterbi Decoder.

B. Synthesis Result

Timing Report:

Minimum period: 5.970ns (Maximum Frequency: 167.504MHz)

Minimum input arrival time before clock: 6.737ns

Maximum output required time after clock: 4.571ns

Maximum combinational path delay: No path found 6.2.2

Area report

Area Report:

Device utilization summary:

Selected Device: 3s500efg320-4

Number of Slices: 27 out of 4656 0%

Number of Slice Flip Flops: 19 out of 9312 0%

Number of 4 input LUTs: 45 out of 9312 0%

Number of IOs: 15

Number of bonded IOBs: 15 out of 232 6%

Number of GCLKs: 1 out of 24 4%

C. RTL Diagram

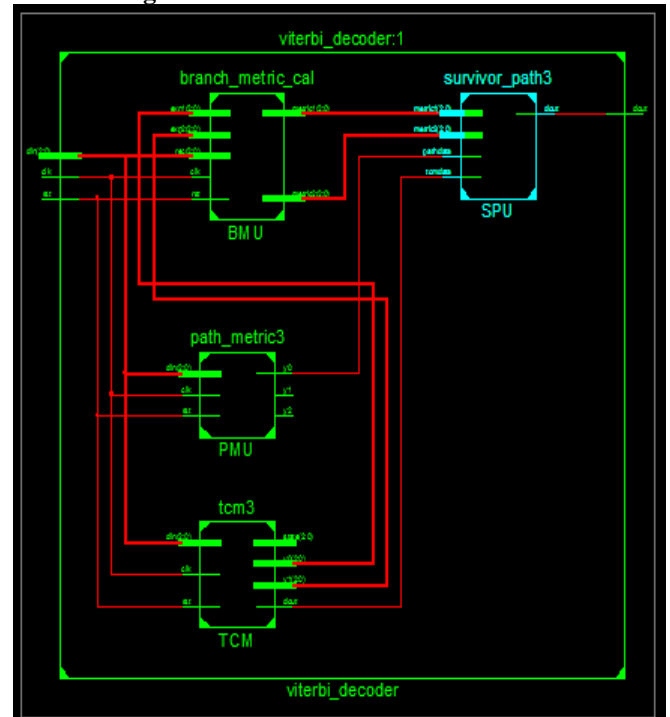


Fig.7. Viterbi Decoder.

VII. CONCLUSION

In this paper, we have presented the design and implementation of the Convolutional encoder and Viterbi decoder for $k/n=1/3$. 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. The Viterbi decoder consists of major blocks of Trellis coded modulation (TCM), Branch metric unit (BMU), path metric unit (PMU), survivor path unit (SPU). This design has been simulated and synthesized using XILINX-ISE 13.2 for the constraint length of $K=7$ and code rate of $1/2$ input sequence. It is implemented in FPGA Spartan 6 Board.

Vinoth, "Implementation of Convolutional Encoder and Viterbi Decoder using Verilog Hdl" in IEEE transactions On inform theory, 2011.

VIII. 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," iee 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.
- [10] G. Davis Forney Jr., "The Viterbi algorithm," Proc. IEEE, vol. 61, pp. 268-278, March 1973.
- [11] 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.
- [12] M. Kling, "Channel Coding Application for CDMA2000 implemented in a FPGA with a Soft Processor Core," Linköping University, Department of Electrical Engineering, 2005.
- [13] R. W. Hamming, "Error detecting and correcting codes," Bell Sys. Tech.J., vol. 29, pp. 147-160, 1950.
- [14] V.Kavinilavu1, S. Salivahanan, V. S. Kanchana Bhaaskaran2, Samiappa Sakthi kumaran, B. Brindha and C.