

## FPGA Implementation of Efficient Viterbi Decoder for Multi-Carrier Systems

K. Rajendar<sup>1</sup>, K. Bapayya<sup>2</sup>

<sup>1</sup>M.Tech., VLSI-SD Student, ECE Dept., CVR College of Engineering, Hyderabad, India

<sup>2</sup>Asst. Prof., ECE Dept., CVR College of Engineering, Hyderabad, India

**Abstract:** In this paper, we concern with designing and implementing a Convolutional encoder and Adaptive Viterbi Decoder (AVD) which are the essential blocks in digital communication system using FPGA technology. Convolutional coding is a coding scheme used in communication systems for error correction employed in applications like deep space communications and wireless communications. It provides an alternative approach to block codes for transmission over a noisy channel. The block codes can be applied only for the blocks of data where as the Convolutional coding has an advantage that it can be applied to both continuous data stream and blocks of data. The Viterbi decoder with PNP (Permutation Network based Path History) management unit which is a special path management unit for faster decoding speed with less routing area. The proposed architecture can be realized by an Adaptive Viterbi Decoder having constraint length,  $K$  of 3 and a code rate ( $k/n$ ) of  $1/2$  using Verilog HDL. Simulation is done using Xilinx ISE 12.4i design software and it is targeted into Xilinx Virtex-5, XC5VLX110T FPGA.

**Keywords:** Convolutional Encoder, Viterbi Decoder, Trellis, PNP Unit, Virtex-5, Verilog HDL, FPGA.

### I. Introduction

Nowadays most of the digital communication systems are employed with Convolutionally encoded data in channel to compensate Additive White Gaussian Noise (AWGN), Fading of the channel, Quantization noise and other data degradation effects. Error detection and error correction is important for reliable communication; error detection techniques are much simpler than Forward Error Correction (FEC). But error detection techniques have certain disadvantages. Error detection pre-supposes the existence of an Automatic Repeat Request (ARQ) feature which acknowledges for the retransmission of data blocks, segments or packets in which errors have been detected.

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. 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 Viterbi Algorithm (VA) was proposed in 1967 by Andrew Viterbi 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. Adaptive Viterbi decoder is very efficient and robust. The main advantage of Adaptive Viterbi Decoder is it has fixed decoding time and also it suites for hardware decoding implementation. The implementation requires the exponential increase in the area and power consumption to achieve increased decoding accuracy.

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 especially for Multi-Carrier Systems like Orthogonal Frequency Division Multiplexing (OFDM).

The general block diagram of Viterbi decoder communication system is shown in Figure 1.

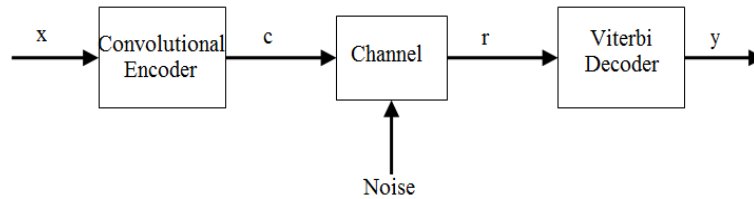


Figure 1: Block diagram of Viterbi decoder communication system

The remainders of this paper are organized as follows. In section II, The important terms and its definitions used in this paper are discussed. In section III, The analysis and FPGA design of (2,1,3) Convolutional encoder are presented. Section IV discusses about the Viterbi Decoder. Section V shows the Results and its discussions based on Xilinx ISE (Integrated Software Environment) 12.4 version tool. In Section VI, The hardware implementation of Convolutional encoder with Viterbi decoder on Xilinx Virtex-5, XUPV5LX110 FPGA board. The paper is concluded in Section VII.

### II. Important Terms And Its Definitions

The following terms are important 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( $n$ ) which corresponds not only current input bit, but also previous ( $K-1$ ) ones.
- 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

The Convolutional encoder is basically a Finite State Machine (FSM). It converts the single bit input into two or more bits for every clock pulse according to the generator polynomial. The generator polynomial gives the connections of the encoder to the modulo-2 adder. In the generator polynomial logic ‘1’ indicates the connection between the stages and logic ‘0’ indicates no connection between the stages. The convolutional encoder can be represented by  $(n,k,K)$  where ‘ $k$ ’ is the number of input bits , ‘ $n$ ’ is the number of output bits and ‘ $K$ ’ is the constraint length that depends up on the number of flip flops used for convolutional encoder. The code rate of a convolutional encoder is defined as ratio of number of input bits to number of output bits. The convolutional encoder with a code rate of  $1/2$  and Constraint length of 3 is shown in Fig. 2.

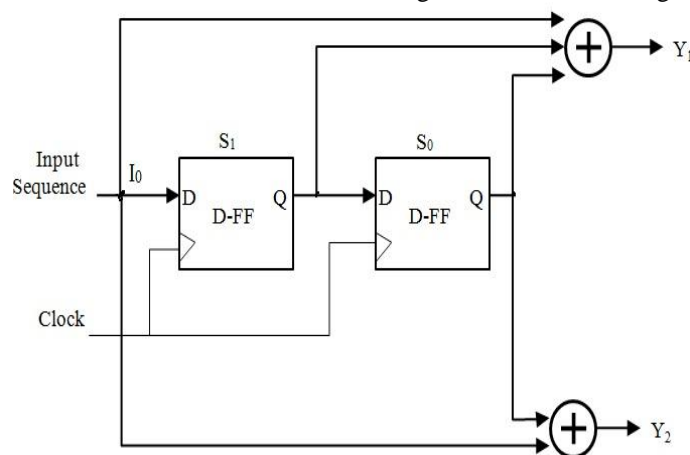


Figure 2: (2,1,3) Convolutional encoder

The Convolutional encoder can be represented by using state table, state diagram and trellis diagram. The state is defined by using the shift register contents of encoder. In state table output symbol is a function of input symbol and state. The state diagram shows the transition between different states. The Trellis diagram is the description of state diagram of encoder by a time line i.e. to represent each time unit with a separate state diagram. The state diagram and the trellis diagram of  $\frac{1}{2}$  rate convolutional encoder in Figure 2 is shown in Figure 3(a) and 3(b) respectively.

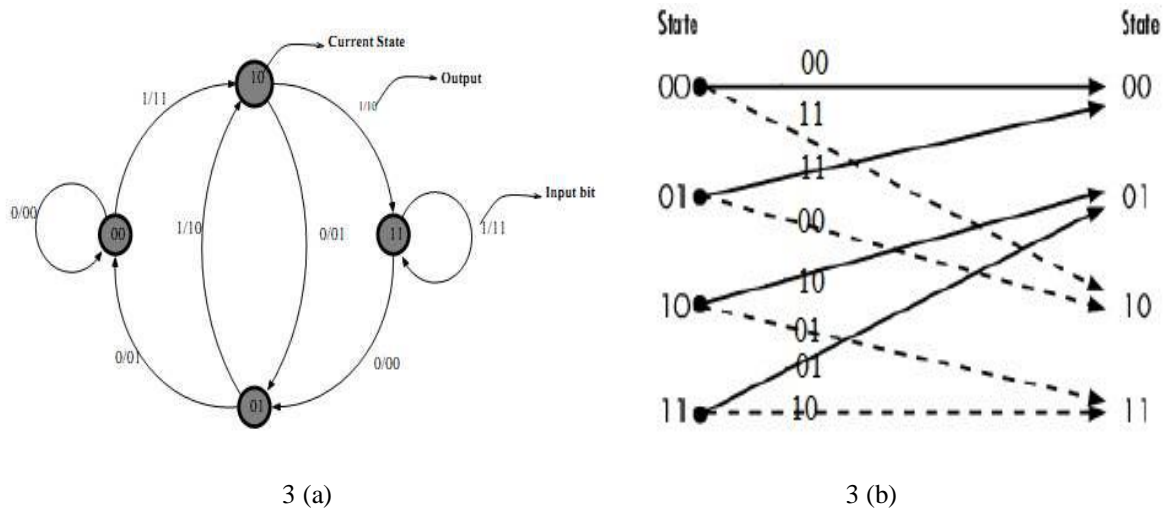


Figure 3: (a) State diagram (b) Trellis diagram

The Trellis diagram is used to find out accumulated distances (called as path metric in viterbi decoder) from the received sequence to get the same transmitted sequence.

#### IV. Viterbi Decoder

When a sequence of data is received from the channel, it is required to estimate the original sequence that has been sent. The process of identifying original message sequence from the received data can be done using the diagram called "trellis". A Viterbi decoder uses the Viterbi algorithm for decoding a bit stream that has been encoded using Forward Error Correction based on a convolutional code. Figure 4 shows the block diagram of Viterbi decoder.

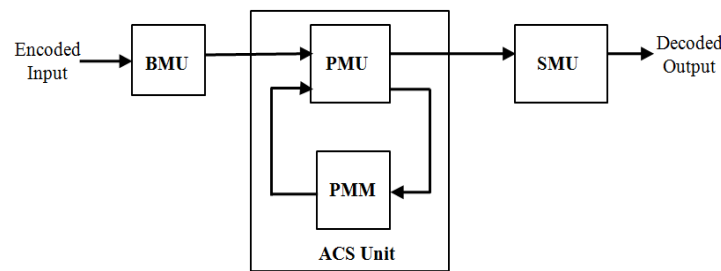


Figure 4: Block diagram of Viterbi decoder

The Viterbi Decoder consists of following functional parts

- a). Branch Metric Unit (BMU)
- b). Add Compare Select Unit (ACSU)
- c). Survivor Memory Management Unit (SMU).

##### a) Branch Metric Unit:

The first unit is called branch metric unit, 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. It is typically a smallest unit of the Viterbi decoder. Its complexity increases exponentially with reciprocal of the coding rate. It is non critical block with respect to area and throughput.

In the hard-decision decoding, the path through trellis is determined by using Hamming distance measure. Thus, most optimal path through trellis is path with minimum Hamming distance. The Hamming distance can be defined as the number of bits that are different between the observed symbol at decoder and the

sent symbol from encoder. Furthermore, the hard decision decoding applies one bit quantization on the received bits.

The Soft-decision decoding is applied for maximum likelihood decoding, when the data is transmitted over the Gaussian channel. On the contrary to hard decision decoding, the soft-decision decoding uses multi-bit quantization for the received bits, and Euclidean distance as a distance measure instead of hamming distance. A 3-bit quantization results in an 8-array output. An implementation of the BMU block is shown in Figure 5.

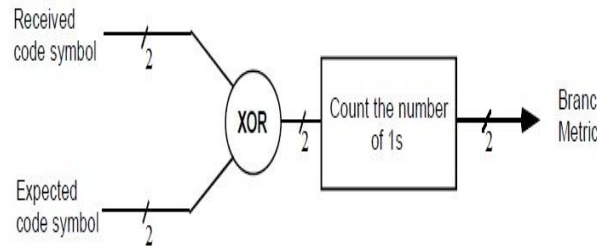


Figure 5: Branch metric computation block

**b) Add Compare and Select Unit:**

The hardware architecture of the ACS module is shown in Figure 6. Path Metric (PM) of the node/state is found by adding the partial path metric from the previous stage and the present stage branch metrics. Since there are two possible ways to reach any node/state two path metrics are obtained. These two are compared to select the one with the least path metric.

The path with the better metric is chosen and stored as the new path metric for current state, while generating a decision bit.

Mathematically,

$$\text{If } BM(i;p) + PM(i;p) < BM(j;p) + PM(j;p) \\ \text{Then } Dec(p)=0, PM(p)=PM(i)+BM(i;p)$$

$$\text{Else} \\ Dec(p)=1, PM(p)=PM(j)+BM(j;p)$$

Where BM is Branch Metric, PM is Path Metric and Dec is Decision bit.

The decision bit indicates what branch was chosen. Because each state can be achieved from the earlier stage, the decision value is represented by one bit. If the bit is '1' the path selected is coming from the lower state from those two possible states in the trellis diagram, and if the decision bit is '0' the path selected is coming from the upper state. As the ACS unit needs the results from the calculations of the previous steps, it forms a feedback loop with the path metric memory unit, where the results are stored.

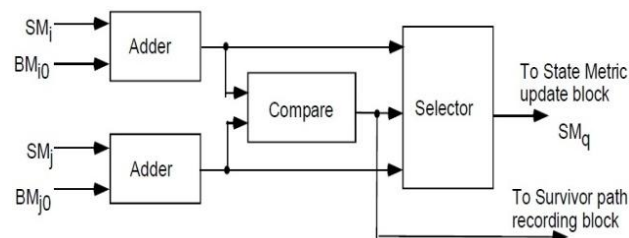


Figure 6: ACS (Add Compare select) module

**c) Survivor Memory management Unit:**

The Survivor Memory management Unit (SMU) stores the decisions of the ACS unit and uses them to compute the decoded output. The Trace-Back (TB) technique and the Register-Exchange (RE) 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 (PNPH) 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 regardless of the constraint length.

**Permutation network based path history (PNPH) unit:**

The Permutation Network based Path History (PNPH) unit for a 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. 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.

Thus, each registers demultiplexer pair determines the part of the survivor path associated its corresponding state node. 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 (2, 1, 3) shown in Figure 7.

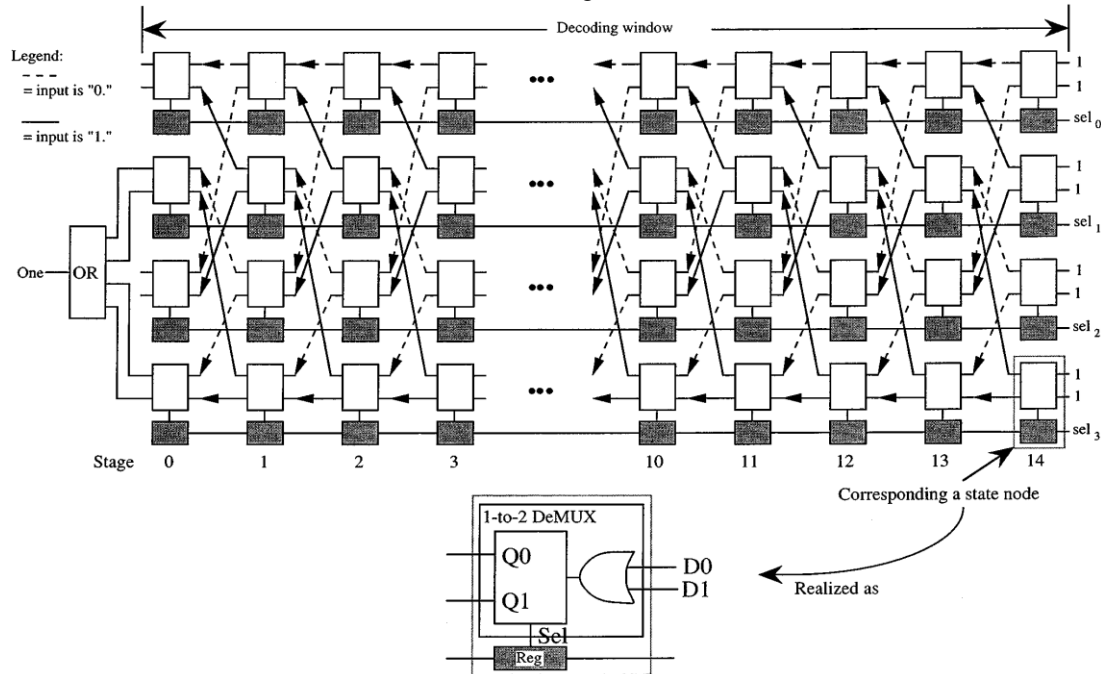


Figure 7: PNPH Unit for the convolutional code (2,1,3)

**V. Results And Discussions**

**1). RTL Schematic:**

The RTL schematic of Convolutional encoder with Viterbi decoder is shown in Figure 8.

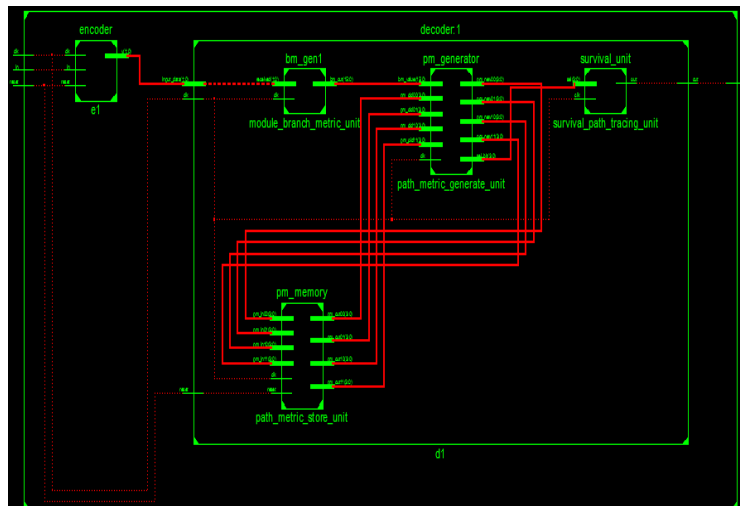


Figure 8: RTL schematic of Convolutional encoder with Viterbi Decoder

2). **Simulation Results:**

i) **Convolutional Encoder:**

The simulation results of Convolutional encoder with constraint length of  $K=3$  and code rate of  $r=1/2$  is shown in Figure 9.

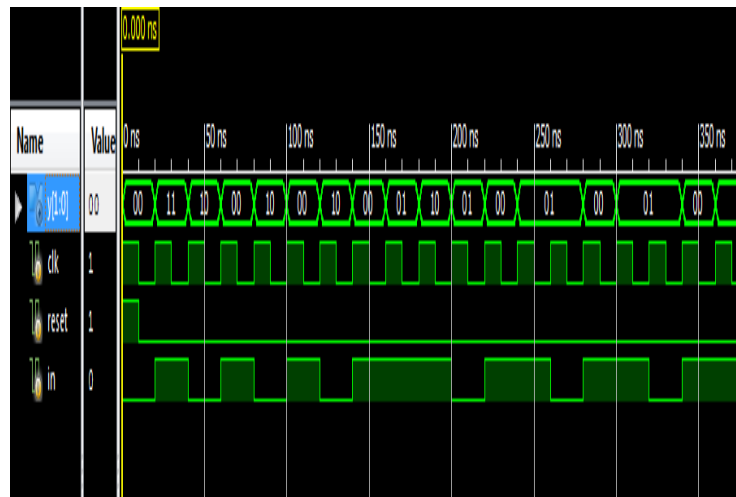


Figure 9: Simulation results of Convolutional encoder

ii) **Viterbi Decoder:**

The simulation results of Viterbi decoder is shown in Figure 10.

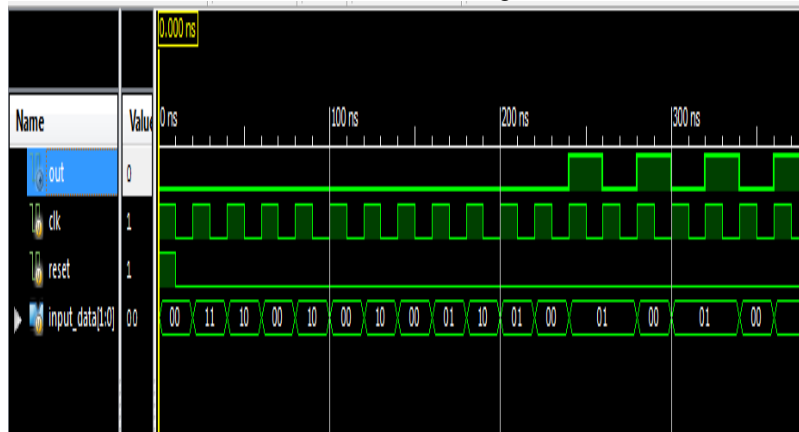


Figure 10: Simulation results of Viterbi Decoder

The simulation results of overall Convolutional encoder with Viterbi Decoder is shown in Figure 11.

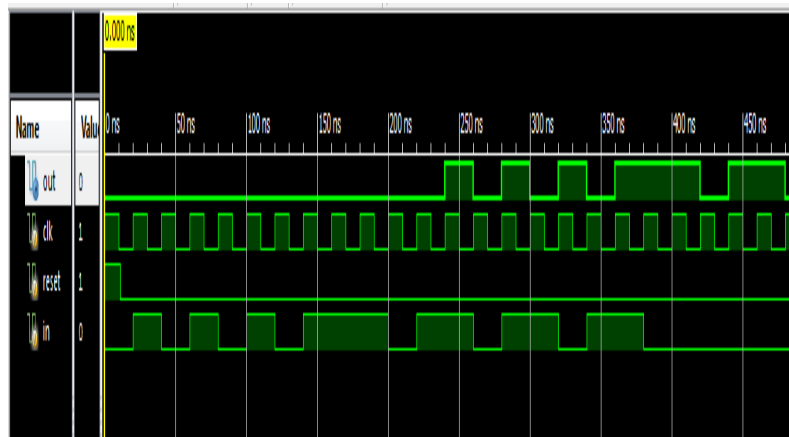


Figure 11: Simulation results of overall Convolutional encoder with Viterbi Decoder

The simulation results of overall Convolutional encoder and Viterbi Decoder with noise is shown in Figure 12.

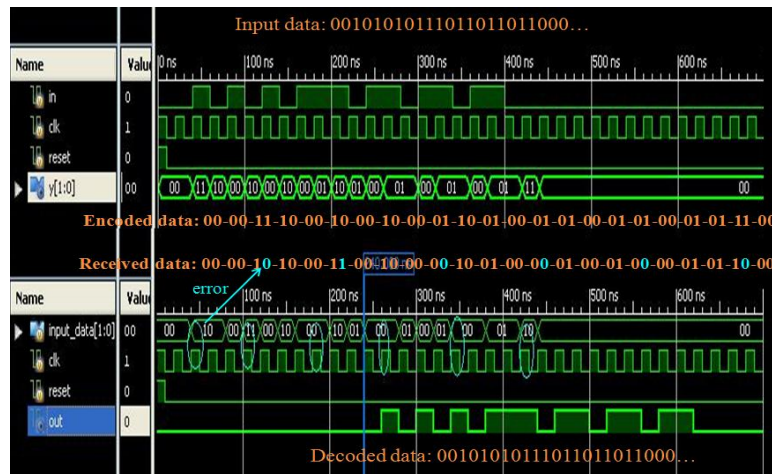


Figure 12: Simulation results of Viterbi Decoder with noise

### VI. FPGA Prototyping

The design of Convolutional encoder and Viterbi decoder is verified on Xilinx Virtex-5, XUPV5LX110T FPGA board by using Xilinx iMPACT device configuration tool. The output of Convolutional encoder with Viterbi decoder on Xilinx Virtex-5, XUPV5LX110T FPGA board with bouncing pattern of LEDs indicating respective outputs is also shown in Figure 13.

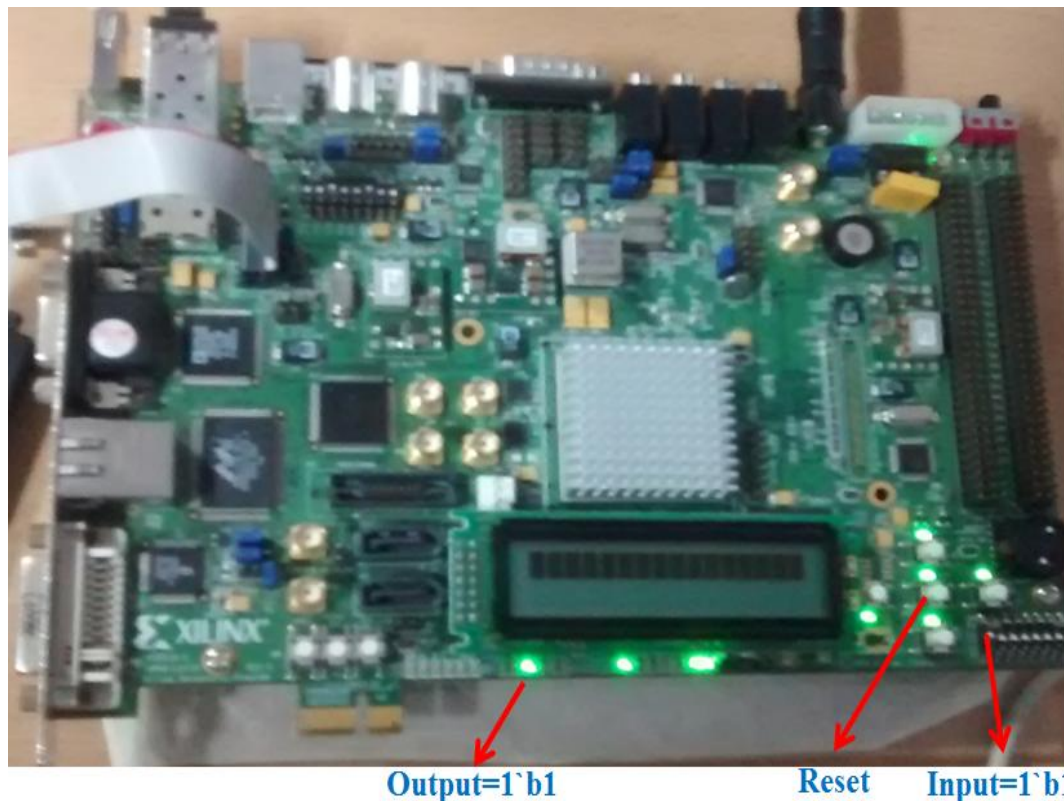


Figure 13: Output of Convolutional encoder with Viterbi decoder on Xilinx Virtex-5, XUPV5LX110T FPGA board

### VII. Conclusion

In this paper, we have presented the design and implementation of the Convolutional encoder and Viterbi Decoder with constraint length of  $K=3$  and code rate of  $r=1/2$ .

The Convolutional encoder and Viterbi decoder with PNPH unit is successfully designed in Xilinx ISE Design suite 12.4 platform with Verilog HDL. The design is simulated for functionality by using Xilinx ISE simulator tool, and Implemented on Xilinx Virtex-5, XUPV5LX110T FPGA board. The synthesized Convolutional encoder with Viterbi decoder has 96 slice LUT's, 78 slice registers and 1-buffer. The timing analysis results show that the critical path is 6.016 ns, i.e. the maximum clock frequency of 166.223 MHz.

As an extension to this work, we can implement Folded PNPH unit to improve the decoding speed and also we can think of reconfigurable architecture for implementing Convolutional encoders with different code rates simultaneously.

#### REFERENCES

- [1] Ming-Bo Lin, "New path history management circuits for viterbi decoders," in IEEE transactions on communications, vol 48, no.10, october 2000.
- [2] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," IEEE Trans. on Information Theory, Vol. IT-13, pp. 260-269, April 1967.
- [3] Anh Dinh, Ralph mason and Joe toth, "High speed v.32 Trellis encoder & decoder implementation using FPGA," in IEEE transactions on communications, 1999.
- [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] 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.
- [6] M. Kling, "Channel Coding Application for CDMA2000 implemented in a FPGA with a Soft Processor Core," Linköping University, Department of Electrical Engineering, 2005.
- [7] W. Chen, "RTL Implementation of Viterbi Decoder," Linköping University, Department of Electrical Engineering, June 2006.
- [8] V.Kavinilavu1, S. Salivahanan, V. S. Kanchana Bhaaskaran2, Samiappa Sakthikumaran, B. Brindha and C. Vinoth, "Implementation of Convolutional Encoder and Viterbi Decoder using Verilog Hdl" in IEEE tran. On inform theory, 2011.
- [9] Hema.S, Suresh Babu.V and Ramesh.P, "FPGA Implementation of Viterbi decoder", proceedings of the 6<sup>th</sup> WSEAS ICEHWOC, Feb. 2007.
- [10] 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.
- [11] B. Sklar, "Digital Communications, Fundamentals and Applications", Second Edition, New Delhi, Pearson Education, 2004.
- [12] Chip fleming, "A tutorial on Convolutional coding with viterbi decoding".