

An Efficient FPGA Implementation of Convolutional Encoder and Viterbi Decoder for DSP Applications

Ranjitha S, Divya Preethi K, Megha K, Students, T Anusha Lalitha,
Assistant Professor
Department of Telecommunication Engineering
BMS College of Engineering,
Bangalore, India

Abstract— In today's scenario of ever increasing usage of wireless devices such as mobile phones and broadband modems, it is important to have a robust error correction codec. Wireless communication systems rely heavily on forward error correction techniques for their proper functioning, thus sending and receiving information with minimal or no error, while utilizing the available bandwidth is a major design concern. Design requirements for modern digital wireless communication systems include high throughput, low power consumption and minimized physical size. The most important aspect dealt in this paper is the implementation of Convolution encoder and Viterbi decoder on FPGA platform by reducing the area and henceforth improving the speed performances. In general digital/wireless communication systems were designed on Digital Signal Processors (DSPs). With the recent trends, Field Programmable Gate Array (FPGA) platform can be summarized as one of the most suitable platform for DSP applications.

In this paper, we present a Convolution encoder and a Viterbi decoder with a constraint length of 3 and code rate of $\frac{1}{2}$. This is realized using Verilog HDL. Simulation and functional verification of the design is performed using ModelSim 10.1b and synthesis using Xilinx ISE. The design is implemented and tested on Xilinx Spartan 3 FPGA and the area utilization of FPGA is derived from the synthesis report.

Keywords— Convolution encoder, Viterbi decoder, Trellis structure, Verilog HDL, FPGA.

I. INTRODUCTION

Convolutional encoding with Viterbi decoding is a dominant method for forward error detection and correction. This method has been widely installed in many wireless communication systems to increase the limited capacity of the communication channels [1]. Convolutional codes are being used extensively in several applications in order to achieve reliable data transfer, including digital video, radio, mobile communication and satellite communications [6]. The Viterbi decoder uses the Viterbi algorithm which fundamentally performs maximum likelihood decoding and this technique reduces the computational complexity by using trellis structure [8]. Viterbi decoder has found universal application in decoding the convolutional codes in both CDMA and GSM digital cellular systems, dial-up modems, satellite, deep-space communications, IEEE 802.11 a/g, WiMAX, DAB/DVB, WCDMA. For all the above-mentioned applications, Field Programmable Gate Array (FPGA) implementation forms the suitable platform.

Today FPGAs play an important role in a wide range of Digital Signal Processing applications. FPGAs offer an opportunity to accelerate the digital signal processing application up to 1000 times over a traditional DSP microprocessor. In recent years, FPGAs are widely used as signal processing appliances and sometimes in union with a processor chip.

Therefore after knowing the advantages of Convolutional encoder and its matching Viterbi decoder and also the FPGAs, we have designed and implemented the former and the latter on FPGA [7]. In this project we have concentrated on implementing both Convolutional encoder and Viterbi decoder on Xilinx Spartan 3 by decreasing the area and improving the speed performances.

II. CONVOLUTIONAL ENCODER

The Convolution encoder consists of shift registers and modulo-2 adders connected to some of the shift registers. The performance of convolution code depends on two parameters: coding rate (r) and constraint length (K) [4]. The input sequence is fed to the K -stage shift registers and output data is calculated using the contents of K -stage shift registers.

The encoder has 'n' modulo-2 adders and 'n' generator polynomials one for each adder. This process doubles the number of input bits at the output. For example, a 4-bit input is converted into an 8-bit output, 8-bit input into a 16-bit output and so on.

The input bits are stored in the fixed length shift registers and they are combined with the help of mod-2 adders. The generator polynomials determine the encoding process. This operation is equivalent to binary convolution and hence it is called "Convolution coding" [5].

A Convolution Encoder accepts an input stream of message and generates encoded output streams to be transmitted. In this process, for one input bit the encoder generates more than one output bits and these redundant symbols in output bit pattern makes the transmitted data more immune to the noise in the channel. The redundant bits help to detect and correct the errors in the received pattern.

Fig. 1 shows a (2, 1, 2) Convolution encoder, with a code rate of 1/2 and a constraint length of 3. The 1/2 code rate means each bit entering the encoder results in 2 bits leaving the encoder.

Hence, we have the following parameters:-

m=2: Two D-flip flops in the encoder

n=2: 2 MOD-2 adders (XOR gates) used

k=1: only 1 bit is clocked in and out of the register each time

K=3: there are 3 shift register stages

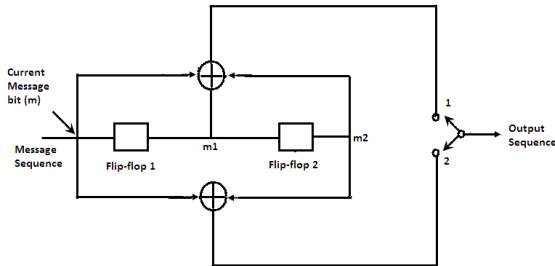


Fig. 1 Convolutional Encoder with k=1, n=2 and r=1/2

A. State Diagram Representation

Fig. 2 shows the state diagram representation of a convolutional encoder. The state diagram illustrates the state information of a Convolutional encoder. The state information of a Convolutional encoder is stored in the shift registers.

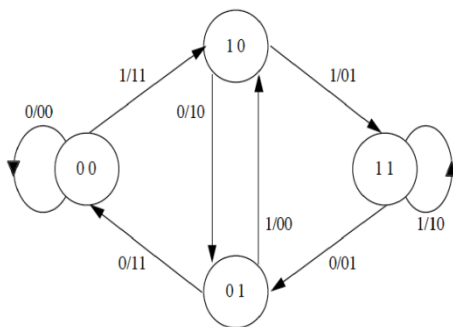


Fig. 2 . State diagram representation of the encoder

III. VITERBI DECODER

Fig. 3 shows the elementary structure of Viterbi decoding system. The basic units of Viterbi decoder are branch metric unit, add compare and select unit and survivor memory management unit [4].

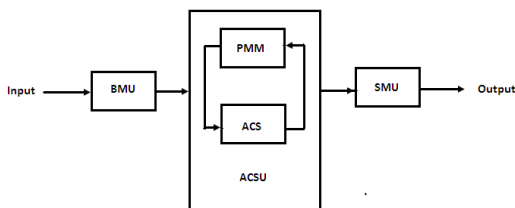


Fig. 3.Viterbi Decoder

A. Basic Units of Viterbi Decoder

Branch Metric Unit (BMU)

It compares the received code sequence with the expected code sequence. It also counts the number of differing bits. The measured value of the BMU can be the Hamming distance in case of the hard decision decoding or the Euclidean distance in case of the soft decision decoding [2].

Path Metric Unit (PMU)

Path metric values of each branch are calculated by adding the branch metrics, associated with a received symbol, to the path metrics from the previous stage of the trellis [8]. The partial path metric is compared by the comparator and selects the smaller value and stores that value as the new path metric for each state.

Survivor Memory Unit (SMU)

This unit will obtain the decoded signal. It consists of two methods namely, trace-back method and register exchange method. The output data and survivor path is obtained by these methods. In this paper we have used trace-back method as it occupies small amount of area.

B. Viterbi Decoder Algorithm

Fig. 4 shows the flow chart of Viterbi decoder. The Viterbi decoding algorithm uses the trellis structure to calculate the path metric value. In this work we have focused on hard decision decoding technique. In hard decision decoding technique, the received signal is either one or zero [3].

- Calculation of branch metrics.
- The shortest path to time n is estimated and the survivor path is updated. This is known as add-compare-select recursion.
- The shortest path pointing to each trellis state is called the survivor path which is obtained by the process called survivor path decode. At last, if all survivor paths are traced back in time, they merge into a unique path, which is the input of convolution encoder.

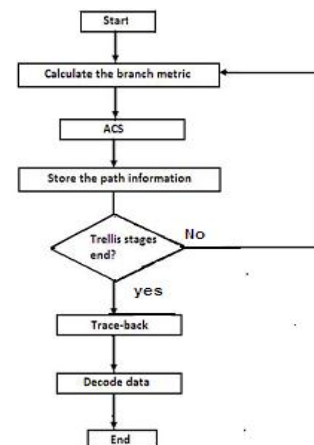


Fig. 4. Viterbi Decoder Flow Chart

C. Trellis Diagram Representation

Fig. 5 shows the trellis diagram. Trellis is a tree like structure with remerging branches. A trellis diagram of a convolution encoder is obtained from its state diagram. In the trellis diagram, each node represents an individual state at a given instant of time and indicate a possible pattern of recently received bits [8]. The transitions which take place between the states are indicated by the branches that are labeled with the corresponding output. The solid lines in the trellis diagram denote an input bit 0 and the dotted lines denote an input bit 1.

Encoded input bits: 11 01 01 11 - - - - input 1

Decoded output bits: 1100 - - - - input 0

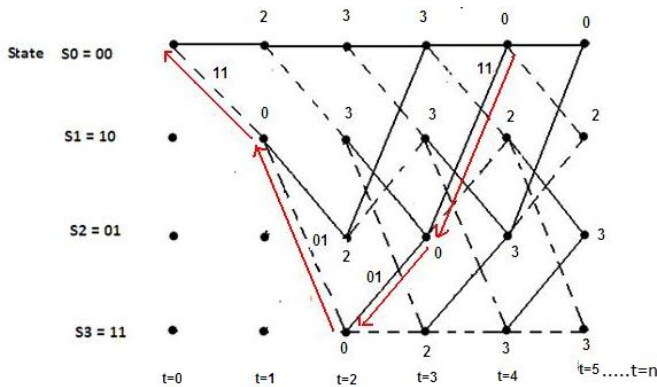


Fig. 5 . Trellis Diagram

IV. IMPLEMENTATION RESULTS

We have designed a Convolutional encoder and a Viterbi decoder using Verilog HDL. Simulation and functional verification of the design is performed using ModelSim 10.1b, and synthesis using Xilinx ISE 14.2. The design is implemented and tested on Xilinx Spartan 3 FPGA and the area utilization of FPGA is derived from the synthesis report. The device utilization summary synthesis report shows that the area utilized by Convolution encoder and Viterbi decoder architecture is less, thus making it an efficient architecture.

A. Simulation Result of Convolution Encoder

Fig. 6 shows the simulation result of Convolution encoder. The Convolutional encoder for the constraint length of K=3 and code rate =1/2 has been developed and synthesis is carried out. Table I shows the device utilization summary and Fig. 7 shows the schematic view of the Convolutional encoder.

Input bits: 1100
Encoded output bits: 11 01 01 11

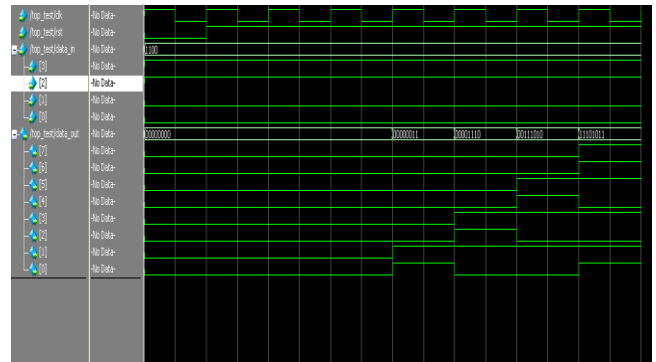


Fig. 6. Simulation result of Convolution encoder

TABLE I: DEVICE UTILIZATION SUMMARY OF THE CONVOLUTION ENCODER

| Device Utilization Summary | | | |
|--|------|-----------|-------------|
| Logic Utilization | Used | Available | Utilization |
| Number of Slice Flip Flops | 24 | 7168 | 1% |
| Number of 4 input LUTs | 20 | 7168 | 1% |
| Number of Occupied Slices | 19 | 3584 | 1% |
| Number of Slices containing only related logic | 19 | 19 | 100% |
| Number of Slices containing unrelated logic | 0 | 19 | 0% |
| Total Number of 4 input LUTs | 20 | 7168 | 1% |
| Number of bonded IOBs | 14 | 141 | 9% |
| IOB Flip Flop | 2 | | |
| Number of BUFGMUXs | 1 | 8 | 12% |

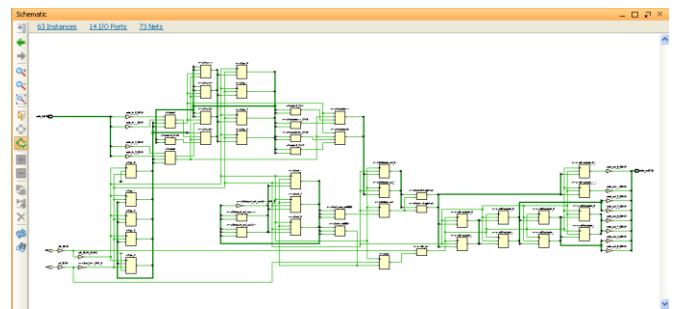


Fig. 7. Schematic view of Convolution encoder

B. Simulation Result of Viterbi Decoder

Fig. 8 shows the simulation result of Viterbi decoder. The Viterbi decoder has been developed and synthesis is carried out. Table II shows the device utilization summary and Fig. 9 shows the schematic view of the Viterbi decoder.

Encoded input bits: 11 01 01 11

Decoded output bits: 1100

V CONCLUSION

A Convolution encoder and Viterbi decoder with Code rate of 1/2 and constraint length of 3 is designed and simulated.

The Convolution encoder and Viterbi decoder are designed as two separate architectures using Verilog HDL simulation and functional verification of the design is performed using ModelSim 10.1b, and synthesis using Xilinx ISE 14.2. The design is implemented and tested on Xilinx Spartan 3 FPGA. An area efficient implementation is achieved at low cost.

Area optimization is a major issue as smaller components allow for better system adaptation, which is an expected feature in present day FPGAs used in Digital Signal Processing applications. We have successfully implemented both convolution encoder and Viterbi decoder by minimizing the area.

ACKNOWLEDGMENT

We are thankful to Dr. N C Shivaprakash, Chief Research Scientist, Indian Institute Of Science, Bangalore for helping and guiding us in completing the project successfully.

We are also thankful to T Anusha Lalitha, Assistant professor, Department of Telecommunication Engineering, BMS College of Engineering, for all the suggestions, being a supportive internal guide throughout and helping us in every possible way.

REFERENCES

- [1] Shin-Pao Cheng, Shi-Yu Huang, "A Low Power SRAM for Viterbi Decoder in Wireless Communication" Consumer Electronics, IEEE Transactions, Volume 54, Issue 2, May 2008, pp.290-295.
- [2] Habib, I. Paker, O. Sawitzki, S., "Design Space Exploration of Hard Decision Viterbi Decoding: Algorithm and VLSI Implementation", IEEE Transaction on very Large Scale Integration (VLSI) Systems, May 2010, pp. 794-807
- [3] FPGA Implementation of Convolutional Encoder and Hard Decision Viterbi Decoder, International Journal of Computer & Communication Technology (IJCCCT), ISSN (ONLINE): 2231 - 0371, ISSN (PRINT): 0975 - 7449, Vol.-3, Issue - 4, 2012.
- [4] Design and implementation of convolution encoder and viterbi decode, IJSR - International Journal of Scientific Research.
- [5] Viterbi, A. J., 1967- "Error bounds for convolutional codes and an asymptotically Optimum decoding algorithm". IEEE Trans. Inform. Theory, vol. 13 no.2, pp.260-269, April 1967.
- [6] Viterbi.A.J, "Convolution codes and their performance in communication systems"- IEEE Transaction on Communications, vol.com- 19, pp. 751 to 771, October 1971
- [7] Sherif Welsen Shaker, Salwa Hussien Elramly and Khaled Ali Shehata, "FPGA Implementation of a Configurable Viterbi Decoder for Software Radio Receiver", Autotestcon, IEEE, July 2009, pp. 140 – 144.
- [8] McEliece, R. J., & Lin, W. (1996). The trellis complexity of Convolutional codes. *IEEE Transactions on Information Theory*, 42(6), 1855–1864.

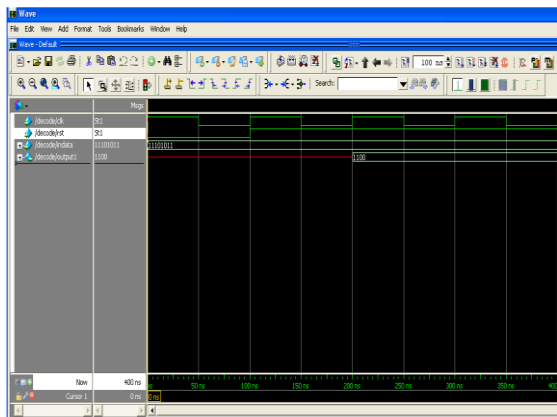


Fig. 8. Simulation result of Viterbi decoder

TABLE II: DEVICE UTILIZATION SUMMARY OF VITERBI DECODER

| Device Utilization Summary | | | |
|--|------|-----------|-------------|
| Logic Utilization | Used | Available | Utilization |
| Total number of Slice Registers | 17 | 7168 | 1% |
| Number used as Flip Flops | 1 | | |
| Number used as Latches | 16 | | |
| Number of 4 input LUTs | 144 | 7168 | 2% |
| Number of Occupied Slices | 87 | 3584 | 2% |
| Number of Slices containing only related logic | 87 | 87 | 100% |
| Number of Slices containing unrelated logic | 0 | 87 | 0% |
| Total Number of 4 input LUTs | 144 | 7168 | 2% |
| Number of bonded IOBs | 14 | 141 | 9% |
| Number of BUFGMUXs | 1 | 8 | 12% |

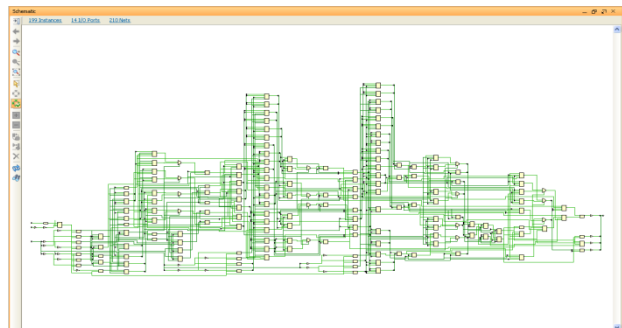


Fig. 9. Schematic view of Viterbi decoder