PERFORMANCE ANALYSIS OF VITERBI DECODER FOR WIRELESS APPLICATIONS

G.Sivasankar and L.Thangarani

Department of Electronics and Communication, Mepco Schlenk Engineering College, Sivakasi, Tamilnadu, India

ABSTRACT

Viterbi decoder is employed in wireless communication to decode the convolutional codes; those codes are used in every robust digital communication systems. Convolutional encoding and viterbi decoding is a powerful method for forward error correction. This paper deals with synthesis and implementation of viterbi decoder with a constraint length of three as well as seven and the code rate of ½ in FPGA (Field Programmable Gate Array). The performance of viterbi decoder is analyzed in terms of resource utilization. The design of viterbi decoder is simulated using Verilog HDL. It is synthesized and implemented using Xilinx 9.1ise and Spartan 3E Kit. It is compatible with many common standards such as 3GPP, IEEE 802.16 and LTE.

KEYWORDS

Convolutional encoder, Constraint length, Code rate, Viterbi decoder, Viterbi algorithm, Hamming distance, Verilog HDL, FPGA.

1. INTRODUCTION

Convolutional coding is a coding scheme used in digital communication systems. It gives an alternative approach to block codes for transmission over a noisy channel. The process of adding the redundant bits or information is known as channel coding. There two types of coding namely block and convolutional code. The block codes can be applied only for the block of data but the convolutional coding can be applied to a continuous data stream as well as to blocks of data [2]. IS-95, a wireless digital cellular standard for CDMA (Code Division Multiple Access), employs convolutional coding. A convolutional code works by adding some structured redundant information to the user's data and then correcting errors using this information.

Viterbi decoding algorithm was developed by Andrew. J. Viterbi in 1967 [1]. It is used as a decoding technique for convolutional codes as well as the bit detection in storage devices in many places. The algorithm operates by forming trellis diagram, which is eventually traced back for decoding the received information. It is also called as dynamic programming which is used for finding the most likely sequence of hidden states, called viterbi path-that results in sequence of observed events. Some of the applications of viterbi decoder include mobile communication, satellite communication, digital cellular telephone etc..,

2. CONVOLUTIONAL ENCODER

Convolutional codes are employed to implement Forward Error Correction (FEC) which is essential component in wireless communication. It achieves error free transmission by adding

some redundant information to the source or input symbols. The convolutional encoder in figure 1 produces two bits of encoded information for each bit of input information, so it is called a rate 1/2 encoder. There are three parameters involved in the convolutional encoder and are n, k and K , where n is number of output bits of the encoder; k is number of input bits of the encoder; K is the constraint length of the encoder. The rate of encoder is defined as k/n. The encoder shown in the figure 1 is a (2, 1, 3) encoder with rate ½. K=m+1, where m is the number of flip-flops of the longest shift register of the encoder. Convolutional encoder increases the length of the message sequence by adding redundant bits in order to increase the likelihood of detecting the transmitted sequence even if errors have occurred during transmission.

Convolutional encoder consists of one or more shift registers and multiple exclusive or EXOR gates. The information bits flows in to the shift register from one end and is shifted out at the other end. EXOR gates are connected to some stages of the shift registers as well as to the current input to generate the output. The block C and D represents the memory elements of the convolutional encoder which is shown in the figure 1. The operation of a convolutional encoder can be easily understood with the aid of a state diagram. Figure 2 represents the state diagram of the convolutional encoder shown in figure 1. Figure 2 depicts state transitions and the corresponding encoded outputs. Here there are two memory elements in the figure 1; so it takes four states. For example, the input information sequence X= {1010} (begin from the all zero state) leads to the state transition sequence S = {10, 01, 10, 01} and produces the output encoded sequence C = {11, 01, 00, 01}. In the figure 2, S0, S1, S2, S3 represents the state values 00,01,10,11 respectively. The generator polynomial used in figure 1 is $1+x+x^2$ (for upper path B) and $1+ x^2$ (for lower path A).

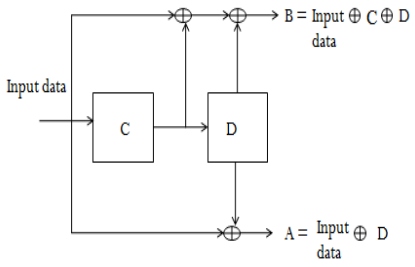


Figure 1. Convolutional Encoder for K=3

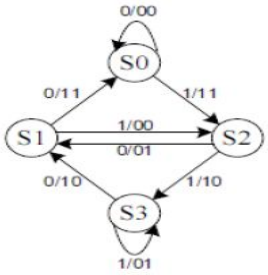


Figure 2. State Diagram of Convolutional Encoder for K=3

This paper also deals with viterbi decoder for K=7 convolutional encoder. The block diagram of convolutional encoder for K= 7 is shown in figure 3. Here the encoder takes 64 states. Generally, the number of states in the encoder is defined as 2^{K-1}. The K denotes the constraint length of the encoder. The generator polynomial used in figure 1 is $1+x+x^2+ x^3+ x^6$ (for upper path A) and $1+ x^2+ x^3+ x^5+ x^6$ (for lower path B).

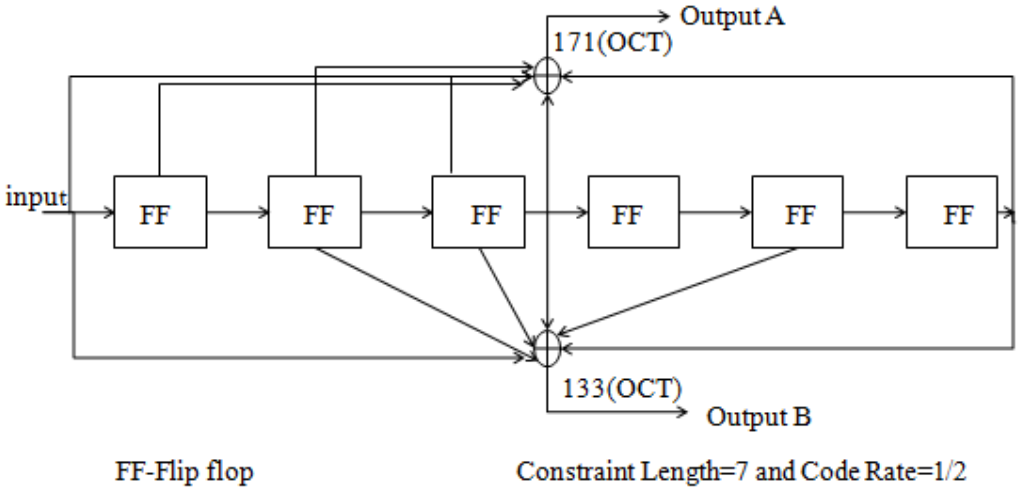


Figure 3. Convolutional Encoder for K=7

3. VITERBI DECODER

Viterbi decoder is most commonly used to resolve convolution codes. This is essential for the purpose of secure transmission of data and its corresponding retrieval during reception. Viterbi decoders also have the property of compressing the number of bits of the data input to half. As a result redundancy in the codes is also reduced; it is more effective and efficient of decoding. Viterbi decoders are based on the basic algorithm which comprises of minimum distance and minimum path calculation and retracing the path. This minimum distance calculation is determined by EX-OR operation and then compared. This can be briefed by means of the block diagram which is shown in figure 4.

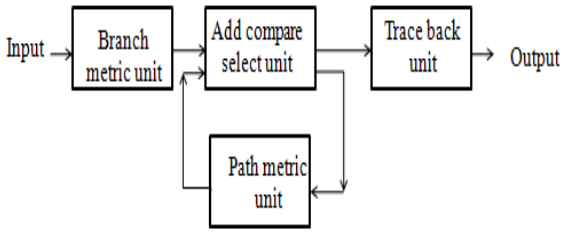


Figure 4. Block Diagram of Viterbi Decoder

To describe the viterbi algorithm of viterbi decoder, one of the inputs of viterbi decoder is taken which are 11110110. The input is taken from in the form of four sets each, consists of 2 bits starting from the Most Significant Bit of the input. The first two bits from the MSB are taken and are EXOR-ed with the values predetermined for the encoder as state0 and the values are obtained. Similarly they are also EXOR-ed with the values of state 1 and the values are also noted and now 2 sets of 4 values each, for state 0 and state 1 is obtained. The corresponding values are compared and the minimum of each of the 2 values is noted down for all the 4 values. Also corresponding position values are marked as 00,01,10,11 respectively for each stage of the trellis diagram. For the minimum path calculation, the value corresponding to that particular minimum path is taken and its state value (state 0 or state 1) is assigned to the minimum path in decimal notation. This is

the function of the Branch Metric Unit. In short it is named as BMU. In general there are two types in viterbi decoding one is soft decision (Euclidean distance is considered for BMU) and another one is hard decision (Hamming distance is considered for BMU). Once the first stage is resolved, the second set of two bits starting from the MSB (11 in our case) is EXOR-ed as explained for the first set. The equivalent value of the minimum distance is taken and is added to the EXOR-ed outputs of the present stage for both value 0 and value 1. They are then compared and minimum distance is once again obtained and its corresponding positions are assigned. This is the function of the Add compare and select unit.

The final step is the trace back procedure, wherein all the values are consolidated to obtain the final output. If the position of minimum value is 00 or 01, a 0 value is obtained in the output. For any other values, 1 is obtained at the output. In this calculation, the minimum value of the last stage is taken and based on its position a 1 or 0 is assigned as one of the output bits. The corresponding path is chosen. The minimum value corresponding to that particular path is considered and once again a position based output bit assignment is made as explained previously, for the stage 3. This is repeated for all the remaining stages. Thus, 4 output bits are obtained. Thus a 4 bit output is derived from an 8 bits input that was given to the viterbi decoder.

4. RESULTS AND DISCUSSION

The Convolutional encoder and Viterbi decoder is simulated and implemented using Verilog HDL and Xilinx Spartan 3E Kit. The Viterbi decoder decodes the original input sequence by using Viterbi algorithm. The proposed design of viterbi decoder occupies fewer amounts of resources when compared with conventional design; it means that it has less hardware complexity. The simulation result, device utilization summary and FPGA Editor of two different encoder and decoder is shown in figures 5,6,7,9,10 and 11 respectively. The power analysis for viterbi decoder is done by Xilinx XPower analyzer which is shown in figure 8. The power consumed by the viterbi decoder is 0.081 Watt which is less compared to previous result [4]. The device utilization summary shows that how much resources like LUT's, flip flops, input and output is utilized by the design of viterbi decoder for the convolutional encoder with the constraint length of three. The Simulation result of convolutional encoder for K=7 is shown in figure 10. The FPGA Editor shows that the amount of area which is occupied by the design of viterbi decoder. The comparative analysis between two constraint lengths is shown in table 1. In this table G1 and G2 represents generator polynomial of convolutional encoder. It denotes the sequence of connections (a one representing a connection and a zero no connection) from the memory elements (flip flops) of a shift register to an output. The comparative analysis result shows that as constraint length increases hardware complexity increases.

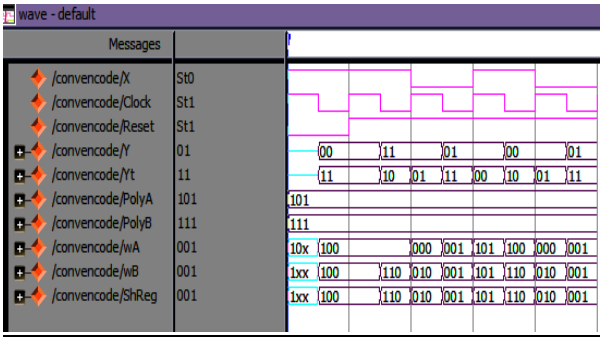


Figure 5. Simulation Result of Convolutional Encoder for K=3

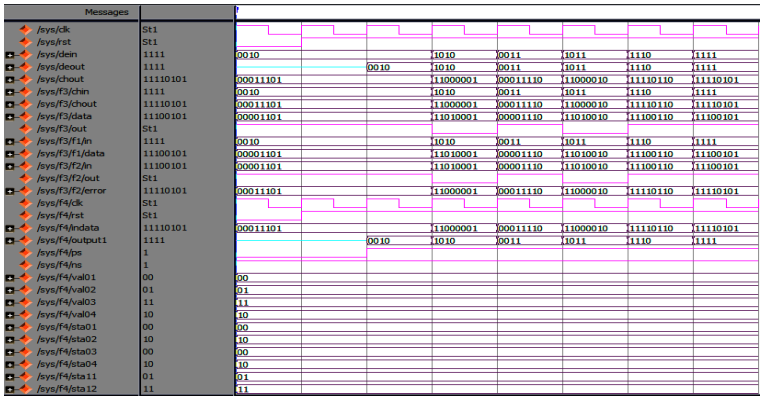
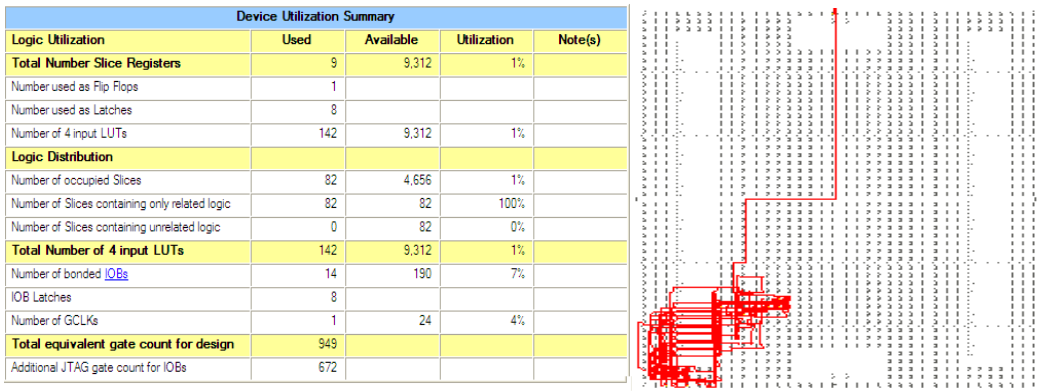


Figure 6. Simulation Result of Viterbi Decoder for K=3



Device Utilization Summary

Logic Utilization	Used	Available	Utilization	Note(s)
Total Number Slice Registers	9	9,312	1%	
Number used as Flip Flops	1			
Number used as Latches	8			
Number of 4 input LUTs	142	9,312	1%	
Logic Distribution				
Number of occupied Slices	82	4,656	1%	
Number of Slices containing only related logic	82	82	100%	
Number of Slices containing unrelated logic	0	82	0%	
Total Number of 4 input LUTs	142	9,312	1%	
Number of bonded IOBs	14	190	7%	
IOB Latches	8			
Number of GCLKs	1	24	4%	
Total equivalent gate count for design	949			
Additional JTAG gate count for IOBs	672			

Figure 7. Device Utilization Summary and FPGA Editor of Viterbi Decoder for K=3

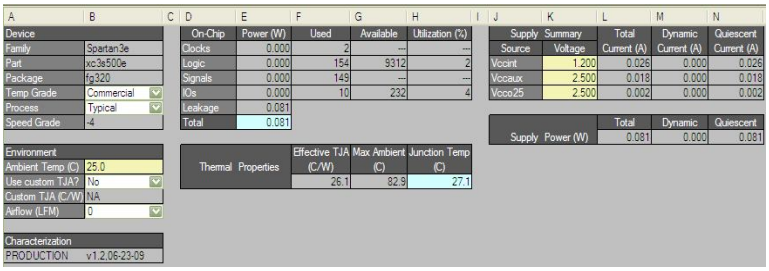


Figure 8. Power Supply Summary of Viterbi Decoder for K=3

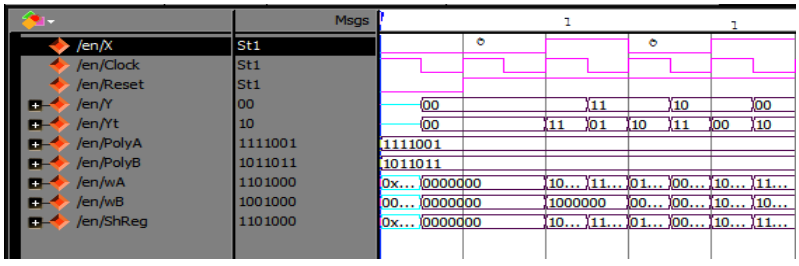


Figure 9. Simulation Result of Convolutional Encoder for K=7

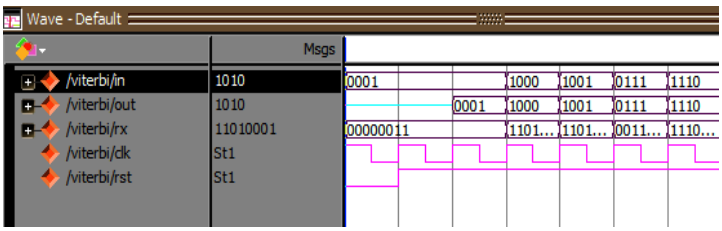


Figure 10. Simulation Result of Viterbi Decoder for K=7

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Total Number Slice Registers	15	9,312	1%
Number used as Flip Flops	1		
Number used as Latches	14		
Number of 4 input LUTs	154	9,312	1%
Number of occupied Slices	84	4,656	1%
Number of Slices containing only related logic	84	84	100%
Number of Slices containing unrelated logic	0	84	0%
Total Number of 4 input LUTs	154	9,312	1%
Number of bonded IOBs	10	190	5%
Number of BUFGMUXs	1	24	4%
Average Fanout of Non-Clock Nets	3.87		

Figure 11. Device Utilization Summary and FPGA Editor of Viterbi Decoder for K=3

Device Utilization summary (xc3s500e-4ft256)		
	Rate ½, K= 3, G1= 101, G2= 111	Rate ½, K= 7, G1= 1111001, G2= 1011011
Logic Utilization	Used	Used
Total no of slice registers	9	15
No of 4 input LUTs	142	154
No of latches	8	14

Table 1. Comparison between two constraint lengths

5. CONCLUSION

The design of viterbi decoder for convolutional encoder is synthesized and implemented on Spartan 3E FPGA. Viterbi decoder can remove potential noise in the incoming stream by decoding it. Even though the viterbi decoder is used in commercial wireless communication purpose, the main complexity of the viterbi decoder, confines its application to convolutional codes with a constraint length K, not exceeding 10.The future work is to improve the speed of the decoding using VLSI techniques like folding, retiming etc.

ACKNOWLEDGEMENTS

I would like to thank my parents who had sacrificed a lot in making me to reach this position. I express my sincere thanks to our Principal Dr. S. Arivazhagan, M.E., Ph.D., for permitting and providing all the facilities to carry out this work in the college campus.

REFERENCES

[1] A.J. Viterbi (1967) "Error bounds for Convolutional Coding and an asymptotically optimum decoding algorithm ", IEEE *Transactions on Inform. Theory*, Vol. 2, pp260-269.

[2] Mr.Vishal, G.Jadhao, Prof. Prafulla & D. Gawade (2012)"Performance analysis of linear block code and convolutional code to study their comparative effectiveness", IOSR Journal of Electrical and Electronics Engineering.

[3] Hiral Pujara, Pankaj Prajapati (2013) "RTL Implementation of Viterbi Decoder using VHDL", IOSR Journal of VLSI and Signal Processing, Vol. 2, pp65-71.

[4] M.Gayathiri et al (2013) "FPGA implementation of high speed and low power viterbi encoder and decoder ", International journal of Engineering and Technology, Vol. 2, pp1315-1320.

Authors

Ms.L.Thangarani is pursuing her master degree in VLSI Design engineering. She has keen interest in VLSI Design Techniques.

Mr.G.Sivasankar is an Assistant Professor in Electronics and Communication Engineering, Mepco Schlenk Engineering College, Sivakasi. He is very much interested in Embedded Sys.