International Journal of Modern Communication Technologies & Research (IJMCTR)
ISSN: 2321-0850, Volume-7, Issue-4, April 2019

Implementation & Simulation of Convolution Encoder & Viterbi Decoder

Vibhuti Sharma, Sunil Sharma

***Abstract*— A forward error correction technique known as convolution coding with Viterbi decoding was explored here. This Viterbi project is part of the baseband Error control coding. Here in this paper, the basic Viterbi encoder &decoder in behavior model was built and simulated. The main aim of this paper is to implement the RTL level model of Viterbi decoder. With the testing results of behavior model, with minimizing the data path, register size and butterflies in the design, we try to achieve a low silicon cost design. The RTL Viterbi decoder model includes the Branch Metric block, the Add-Compare-Select block, the trace-back block, the decoding block and next state block. With all done, we further understand about the Viterbi decoding algorithm and the DSP implementation methods. Present wireless standard such as third generation (3G) systems, GSM, 802.11A, 802.16 utilize some configuration of convolution coding. Convolution encoding with viterbi decoding is a powerful method for forward error correction. The viterbi algorithm, which is the most extensively employed decoding algorithm for convolution codes. The aim of this paper is to design convolution encoder and viterbi decoder with a constraint length of 3 and code rate of 2/3. This is realized using Verilog HDL.It is simulated and synthesized using Modelsim Altera 10.1d.**

***Index Terms*— convolutional encoder, viterbi decoder, look-up table, modelsim 10.1d, viterbi algorithm.**

I. INTRODUCTION

In telecommunication and information theory, forward error correction(FEC) also called channel coding is a system of error control for data transmission, whereby the sender adds systematically generated redundant data to its messages, also known as an error correction code(ECC) [1].

Convolution codes are employed to implement FEC. In telecommunication, a convolution code is a type of error-correcting code in which

Each m-bit information symbol (each m-bit string) to be encoded is transformed into an n-bit symbol, where m/n is the code rate (n>m) and the transformation is a function of the last k information symbols, where k is the constraint length of the code [2]. Convolution codes are used extensively in numerous applications in order to achieve reliable data transfer, including digital video, radio, mobile communication and satellite communication.

A viterbi decoder uses the viterbi algorithm for decoding a bit stream that has been encoded using forward error correction based on a convolution code. There are other algorithms for decoding a convolution encoded stream (for example, the FANO algorithm). The viterbi algorithm is the most resource-consuming, but it does the maximum likelihood decoding [2].

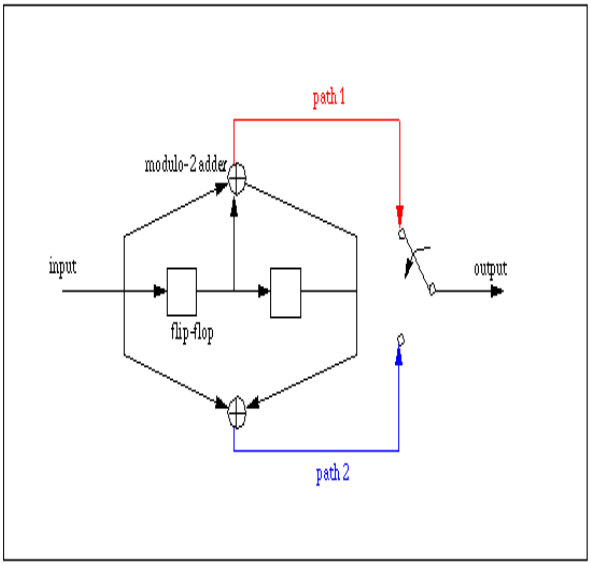

Figure1. Block diagram of convolution encoder

II. PROPOSED WORK & METHODOLOGY

Viterbi Encoder:

Viterbi encoding is widely used for satellite and other noisy communications channels. There are two important components of a channel using Viterbi encoding: the Viterbi encoder (at the transmitter) and the Viterbi decoder (at the receiver). A Viterbi encoder includes extra information in the transmitted signal to reduce the probability of errors in the received signal that may be corrupted by noise. We shall describe an encoder in which every two bits of a data stream are encoded into three bits for transmission. The ratio of input to output information in an encoder is the rateof the encoder; this is a rate 2/3 encoder. The following equations relate the three encoder output bits (Y_{2N} , Y_{1N} , and Y_{0N}) to the two encoder input bits (X_{2N} and X_{1N}) at a time nT:

$Y_{2N} = X_{2N}$, Y1N = X_{1N} xor X_{1N_2}, $Y_{0N} = X_{1N\_1}$

We can write the input bits as a single number. Thus, for example, if X_{2N} = 1 and X_{2N} = 0, we can write X_N = 2. Equation defines a state machine with two memory elements for the two last input values for X_{1N}, X_{1N_1} and X_{1N_2} . These two state variables define four state s: {X_{1N_1}, X_{1N_2} } , with S_0 = { 0, 0}, S_1 = {1, 0}, S_2 = {0, 1}, and S_3 = {1, 1}. The 3-bit output Y_n is a function of the state and current 2-bit input X_n.

Sunil Sharma,Assistant Professor & Head, Department of Electronics & communication Engineering, Pacific University, Udaipur

Vibhuti Sharma, M.Tech. Scholar, Department of Electronics & communication Engineering Pacific University, Udaipur

State table for the rate 2/3 Viterbi encoder.

Present state	Inputs		State variables		Outputs			Next state	
	X_{2N}	X_{1N}	X_{1N_1}	X_{1N_2}	Y_{2N}	Y_{1N} = X_{1N} xor X_{1N_2}	Y_{0N} = X_{1N_1}	{X_{1N_1}, X_{1N_2}}	
S_0	0	0	0	0	0	0	0	00	S_0
S_0	0	1	0	0	0	1	0	10	S_1
S_0	1	0	0	0	1	0	0	00	S_0
S_0	1	1	0	0	1	1	0	10	S_1
S_1	0	0	1	0	0	0	1	01	S_2
S_1	0	1	1	0	0	1	1	11	S_3
S_1	1	0	1	0	1	0	1	01	S_2
S_1	1	1	1	0	1	1	1	11	S_3
S_2	0	0	0	1	0	1	0	00	S_0
S_2	0	1	0	1	0	0	0	10	S_1
S_2	1	0	0	1	1	1	0	00	S_0
S_2	1	1	0	1	1	0	0	10	S_1
S_3	0	0	1	1	0	1	1	01	S_2
S_3	0	1	1	1	0	0	1	11	S_3
S_3	1	0	1	1	1	1	1	01	S_2
S_3	1	1	1	1	1	0	1	11	S_3

Table 1 Viterbi encoder table

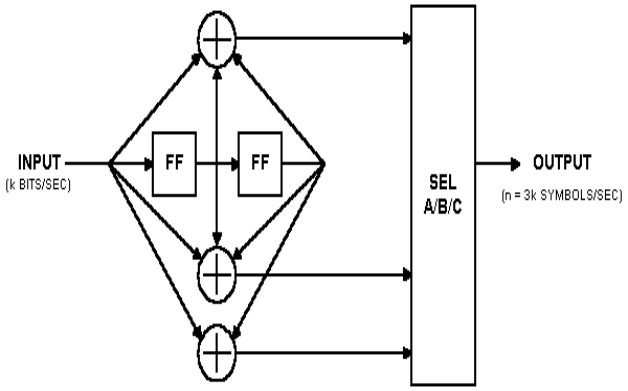

Figure 2. Convolution encoder with three bit output

Viterbi decoding of convolution codes:

There are three methods for decoding the convolution codes. They are as under:

1. Viterbi algorithm
2. Feedback decoding
3. Sequential decoding

The viterbi decoder:

The decoding algorithm uses two metrics: the branch metric (BM) and the path metric (PM). The branch metric is a measure of the "distance" between what was transmitted and was received, and is defined for each arc in the trellis. In hard decision decoding, where we are given a sequence of digitized parity bits, the branch metric is the hamming distance between the expected parity bits and the received ones [10].

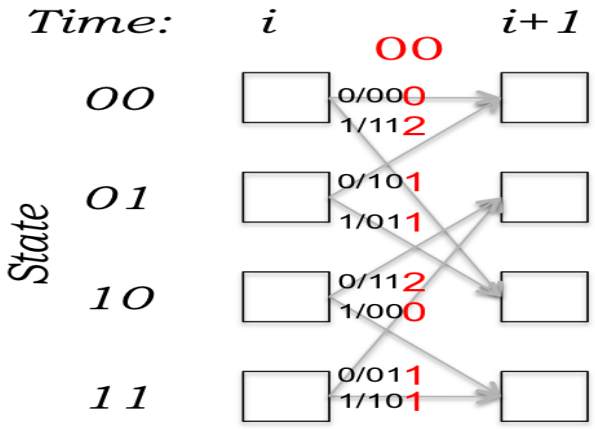

Figure3. The branch metric for hard decision decoding, in this example, the receiver gets the parity bits 00.

Figure 3, where the received bits are 00. For each state transition, the number on the arc shows the branch metric for that transition. Two of the branch metrics are 0, corresponding to the only states and transitions where the corresponding hamming distance is 0. The other non-zero branch metrics correspond to cases when there are bit errors. The path metric is a value associated with a state in the trellis (i.e. a value associated with each node). For hard decision decoding, it corresponds to the hamming distance over the most likely path from the initial state to the current state in the trellis. By "most likely", we mean the path with smallest hamming distance between the initial state and current state, measured over all possible paths between the two states. The path with the smallest hamming distance minimizes the total number of bit errors, and is most likely when the BER is low. The key insight in the viterbi algorithm is that the receiver can compute the path metric for a (state, time) pair incrementally using the path metrics of previously computed states and branch metrics.

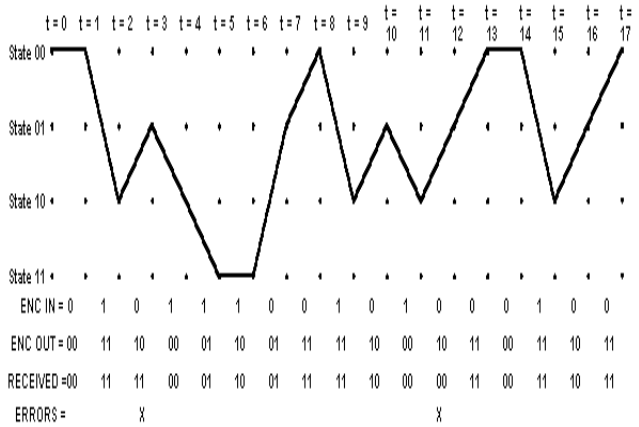

Figure 4. Encoding & Decoding Process

It indicates hamming distance calculation for every possible state transition for given input combinations. The main idea is to follow this table in reverse direction from last time instant to front direction. E.g. at t=17, if state is 00, we will move to minimum distance state at t=16, say for example 01. In this way we will continue reverse tracking up to t=0 point. Now from this process we will be able to decide or identify inputs for state movements, we will take help from table of state transition and table of input /output relation.

Basically we need distance calculation, addition and memory design for this entire process. We have written verilog code for adder, memory design, and distance calculation with the assumption that modulation technique utilized is 8 PSK.

In case of 8 PSK there will be eight different phases for each combination of inputs. Here we have 2/3 rate encoder so Y[0],Y[1],Y[2] are three output bits of encoder , which will be received by decoder front end . These bits will be processed by receiver for distance calculation.

Say for example if encoded signal is at 000 place , we will calculate hamming distance between 000 and all remaining possible combination of three bits , from 001 to 111. This distance will be expressed in terms of angle.

The 2/3 rate encoder is suitable for PSK, where, We have used Modelsim 10.1d platform for simulation. This software is provided by ALTERA for education and research purpose.

III. Simulation Results of Proposed Work

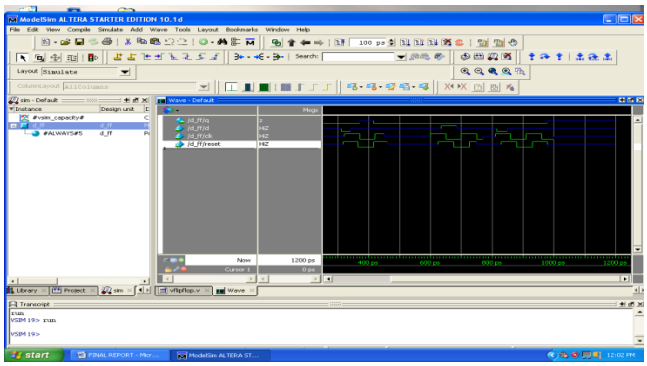

Figure 5. Waveform of D flip-flop

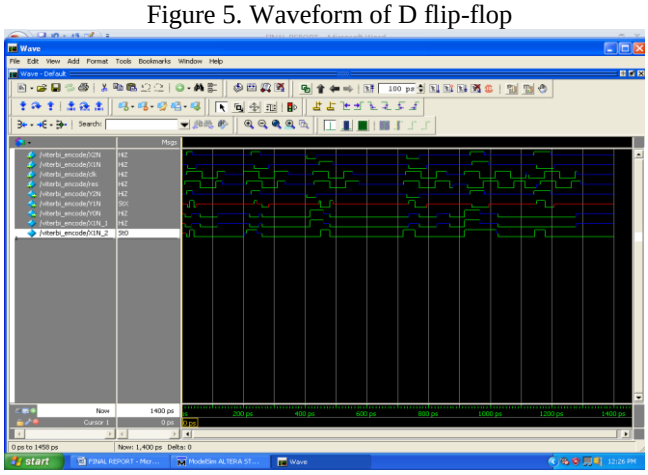

Figure 6. Waveform of 2/3 code rate viterbi encoder

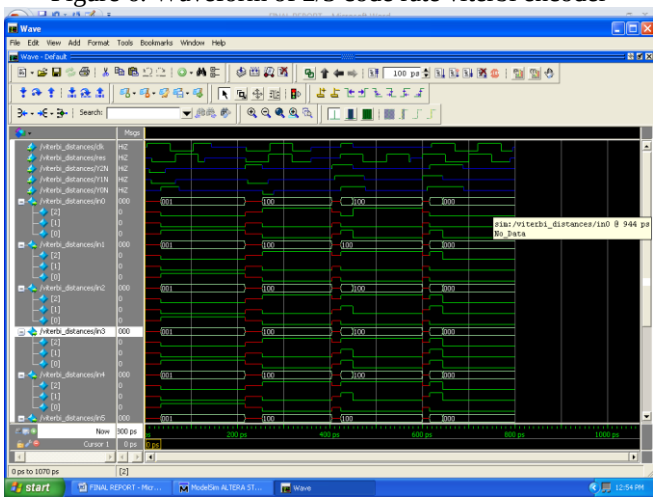

Figure 7. Waveform of Hamming distance

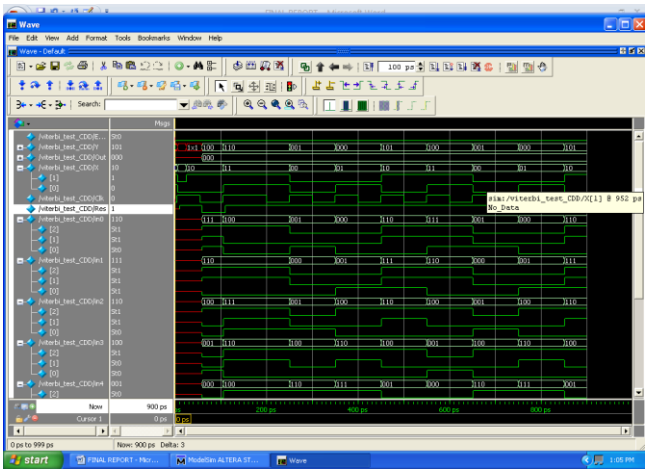

Figure 8. Waveform of Testbench of viterbi encoder & decoder

IV. Conclusion

The goal was to implement convolution encoder and decoder pair in compact verilog style. As verilog implementation works in module form and it is comparatively simple than other Hardware Description language. Second, we wanted to design decoder model by look up table method at receiver end so that, implementation would take less memory space. While tracing path back towards front end, look up method saves lot of time and complexity.The modulation scheme 8 PSK was used here for faster coding with three bit input frame. By building the convolution encoder and Viterbi decoder in the behavior model, the Modelsim simulation results give us a light on its performance.

Our main future task will be to measure & improve the performance of designed encoder –decoder pair. The next level implementation will also consist power saving coding style. Our code is written in specific way to make the design low power design.

Our secondary future goal will be comparative study of power consumption of our code with other rtl code.

References

[1] pravallika.kolakaluri, R.Suryaprakash: HDL Implementation of convolutional encoder and viterbi decoder; july, 2012;

[2] Rohan M. Pednekar, Dayanand B M: Design and Implementation of convolutional encoder and viterbi decoder; 2013;

[3] Swati Gupta, Rajesh Mehra: FPGA Implementation of viterbi decoder using track back architecture; june, 2011;

[4] Mahe Jabeen, Salma Khan: Design of convolutional encoder and Reconfigurable viterbi decoder; (sept 2012);

[5] Samir Palnitkar: Verilog HDL A Guide to Digital Design and Synthesis; sunsoft press(1996);

[6] Chip Fleming: A Tutorial on Convolutional Coding with Viterbi Decoding; Spectrum Application; Jun, 2003;

[7] Dake Liu: Design of Embedded DSP processors; Computer Engineering Division of Electrical Engineering Department, Linköping University; 2004

[8] Stephen B. Wicker: Error Control Systems for Digital Communication and Storage; Prentice Hall; (1995)

[9] John G. Proakis, MasoudSalehi; Contemporary communication systems using MATLAB; Brooks; 2000

[10] Rodger E. Ziemer, Roger L. Peterson: Introduction to digital communication; Prentice-Hall International (UK), cop. 2001

[11] SamirkumarRanpara: On a Viterbi Decoder design for low power dissipation; Virginia Polytechnic Institute and State University; April, 1999;

[12] Simon Haykin and Michael Moher: Modern wireless communications; Pearson Prentice Hall, 2005

[13] Simon Haykin; Digital communications; Wiley, cop. 1988