FPGA Design and Implementation of Convolution Encoder and Viterbi Decoder

Mr.J.Anuj Sai[1], Mr.P.Kiran Kumar[2],
Mr. V.Vamshi Krishna Reddy[3],Mr.L.Shiva Nagender Rao[4]
[1]UG Scholar, Dept of ECE, SNIST, Hyderabad, Email: anujsai96@gmail.com
[2]UG Scholar, Dept of ECE, SNIST, Hyderabad, Email:kiranponugoti32@gmail.com
[3]UG Scholar, Dept of ECE, SNIST, Hyderabad, Email: vuppulavkreddy@gmail.com
[4]UG Scholar, Dept of ECE, SNIST, Hyderabad, Email: shivanag.lopelli@gmail.com

Abstract

Communication system transmits data from source to transmitter through a channel which may be wired or wireless. In this paper let us see about wireless channel communication. Data transmission over wireless channel is affected by attenuation, distortion, interference and noise, which affect the receiver's ability to receive correct information. Convolution encoding with Viterbi decoding is a powerful method for forward error correction. This FEC scheme is an essential component of wireless communication systems. Convolutional codes are employed to implement FEC but the complexity of corresponding decoders elevates exponentially according to the constraint length. The transmitted signal is corrupted mainly by "Additive White Gaussian Noise" (AWGN). The maximum likelihood detection of a digital stream is possible by Viterbi Algorithm. This paper focuses on the realization of Convolutional Encoder and Viterbi Decoder (VD) with a constraint length(K=5) and a code rate (k/n=1/2) using FPGA technology. The performance of the implemented Convolutional Encoder and Viterbi Decoder is realized using Verilog HDL and is simulated by using Xilinx 13.1i in ARTIX-7.

Keywords: FPGA, Convolutional encoder, Viterbi decoder, Verilog HDL, Trace Back (TB) method.

1. Introduction

As there is an increase in use of digital communication, there has been an increased interest in Viterbi decoder design within a single chip. Viterbi decoder with the throughput at the order of Giga-bit per second, without using off-chip processor(s) or memory can be realized using Advanced field programmable Array(FPGA) and well developed electronic design automatic (EDA) tools . The main purpose of this thesis is to use VHDL, Synopsys synthesis and simulation tools to realize a Viterbi decoder having constraint length 5 aiming Xilinx FPGA technology

In information theory and telecommunication, forward error correction (FEC) (also called channel coding) is a process of error detection and coreection for data transmission, here transmitter adds redundant data to its messages, also known as an error-correcting code (ECC).these bits which are added up takes care of error detection at receiver end. FEC gives the receiver an ability to correct errors without needing a reverse channel to request retransmission of data.FEC is therefore applied in situations where retransmissions are relatively costly, or impossible such as when broadcasting to multiple receivers. This mechanism of adding bits for error control is called channel coding. Channel coding is divided into two. They are convolution coding and block coding. In this paper we study about convolution codes.

Convolution codes were first introduced in 1955 by P.Elias. Convolution codes are generally error correcting codes that are used to improve the performance of many digital systems such as digital radio, mobile phones and the Bluetooth implementations. Viterbi decoding was developed by Andrew J. Viterbi in 1967 . The Viterbi algorithm is an optimum decoding technique. Viterbi algorithm is a maximum likelihood algorithm and performs decoding by searching the minimum cost path in a weighted oriented graph, named trellis. The basic building blocks of Viterbi decoder are Branch Metric Unit (BMU), Path Metric Unit (PMU), Add Compare and Select Unit (ACSU) and Survivor Memory Management Unit (SMU).

Convolution coding with Viterbi decoding is a FEC technique is used to that channel which is corrupted by additive white Gaussian noise (AWGN). Generally in audio and video applications, the convolutional codes are used for error detection and correction. 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).

In this paper we shall assume a 5-bit and give as an input to the convolutional encoder rate-1/2 code (two output bits for every input bit). This will yield a 2x5 output matrix, with the extra bits allowing for the correction. This 10 bit output is given as the input to the viterbi decoder which decodes the convolutional codes into original data.

2. Research Method

This architecture mainly consists of three main blocks, inputs are been converted in convolution codes using Convolutional encoder an then transmitted through AWGN Channel , finally at receiver end decoding takes places using Viterbi algorithm.

2.1 Convolutional Encoder

The design of convolution encoder is realized by using a Finite State Machine (FSM). This convolution encoder is made of a fixed number of shift registers.
Convolutional codes are generally represented by the three parameters (n,k,m). where n = number of output bits, k = number of input bits and m = number of memory registers. The quantity k/n called the code rate is a measure of the bandwidth efficiency of the code. These Convolutional codes are employed to implement FEC .It take a single or multi-bit input and generate a matrix of encoded outputs.

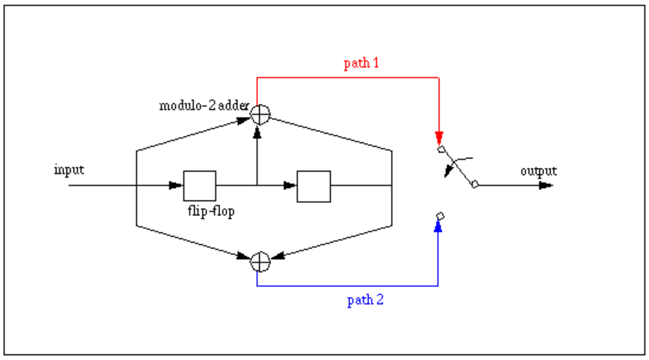


Figure 2.1.1:Convolutional Encoder of code rate ½.

Above shows block diagram of ½ rate convolutional encoder. Using this let us draw truth table.

Trellis Diagram: Trellis diagram can be obtained by using above truth table. Firstly write the all possible present state buffers (C D),followed by next state (C D). Now by referring the above table, for zero input and the buffer values are 00 (C D) the respective next state is 00 (C D). Thus draw a straight line between 00 of present state to 00 of next state, and above the straight line write the output value (A B) in bracket. As like the same for one input, but the line is dotted line.

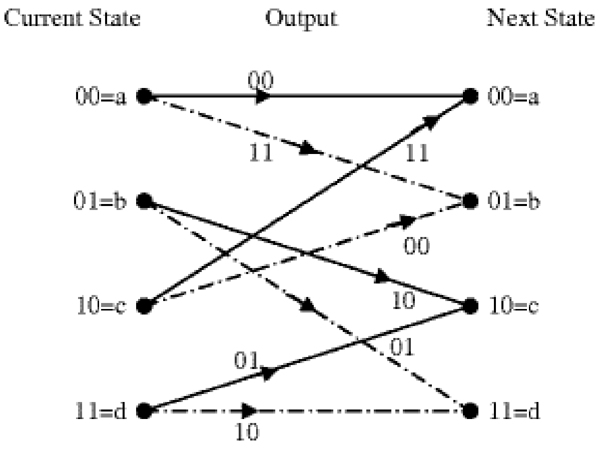


Fig 2.1.2: Trellis diagram

State diagram:

As Convolutional encoding is designed using FSM, State diagram can be obtained.

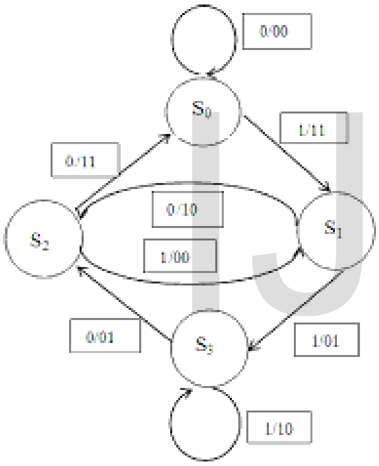


Fig 2.1.3: State diagram

2.2 Design of AWGN Channel

The AWGN Channel block adds white Gaussian noise to a real or complex input signal. When the input signal is real, this block adds real Gaussian noise and produces a real output signal. When the input signal is complex, this block adds complex Gaussian noise and produces a complex output signal. This block inherits its sample time from the input signal.

2.3Viterbi Decoder

Viterbi algorithm:

The Viterbi decoder is operated by using Viterbi decoder. The Viterbi algorithm has been known as a maximum likelihood decoding algorithm for convolutional codes. It uses the trellis diagram to measure path metric values. As number of inputs increases path metric also increases, this eventually increases the complexity and memory problems. The Viterbi algorithm can be classified into hard decision decoding and soft decision decoding. In the hard decision decoding, the path over the trellis is resolved using the Hamming distance measure.

Hamming distance is defined as number of bits present in observed to that of sent data from encoder. For soft decision decoding ML decoding is applied.

The implementation of the VA consists of three parts: branch metric computation, path metric updating, and survivor sequence generation. The path metric computation unit computes a number of recursive equations. The main drawback of these Viterbi Decoders is that they are very expensive in terms of chip area.

Viterbi Decoder:

Viterbi decoder are used to resolve convolution codes. This can secure the data during transmission and also can retrieve it. 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. Hence viterbi decoding is more effective and efficient

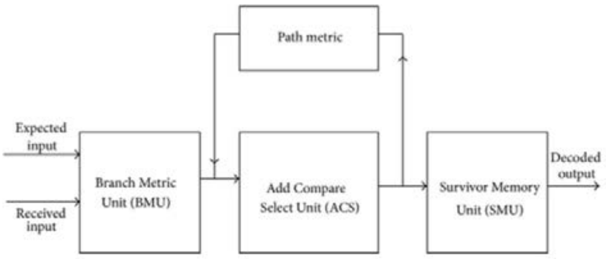
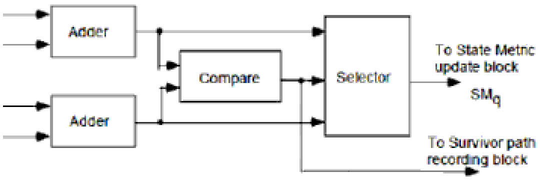


Fig 2.3.1: Viterbi Decoder block diagram

2.3.1 Branch Metric Unit

This is first unit present in decoder. Branch Metric unit compares the received data with the expected code and counts the number of differing bits, which is called Hamming distance. Thus hamming distance is used for branch metric computation.

2.3.2 Add Compare Select


Adder
Adder
Compare
Selector
To State Metric update block
SM_q
To Survivor path recording block

Here in this unit two adders calculate the partial path metric of each branch an sends to comparator, this comparator compares two path metric. Finally selector selects a particular branch. The new partial path metric updates the state metric of state, and the survivor path-recording block records the survivor path.

2.3.3 Path Metric Unit

The path metric unit includes the Add compare and select unit. The partial path metrics are compared by the comparator and branch metric is selected by the selector. That means the selector selects the smaller value. The second unit, called a path metric computation unit, estimates the path metrics of a step by adding the branch metrics, combine with a received symbol, to the path metrics from the earlier stage of the trellis.

2.3.4 Survivor Memory Management Unit

The Survivor Memory Management Unit (SMU) can be designed by using two methods. One is Register Exchange method and the other is Traceback method. The Register Exchange approach assigns a register to each state. The register records the decoded output sequence along the path starting from the initial state to the final state, which is same as the initial state. This method can be implemented by using Euclidean distance.

In the TB method, the storage can be implemented as RAM and is called the path memory. After at minimum L branches have been processed, the trellis connections are recollected in the reverse order and the path is traced back through the trellis diagram. This is implemented by Hamming distance.

3. Results and Analysis

In this section, Convolution encoder, AWGN channel and Viterbi decoder blocks designed using Xilinx synthesis tool are described.

3.1 Encoder Implementation

Coding for this encoder part is done on verilog and encoding is tested for various information message blocks satisfactorily. RTL schematic for encoder block obtained from Xilinx as shown in figure 3.1

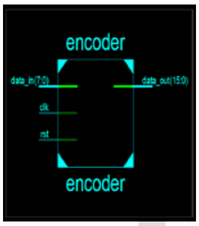


Fig 3.1 Encoder RTL schematic

3.2 AWGN Channel Design

AWGN is often used as a channel model in which the only impairment to communication is a linear addition of wide band or white noise with a constant spectral density and a guassian distribution of noise.

The AWGN channel is a good model for many satellite and deep space communication links. It is not a good model for most terrestrial links because of multipath, terrain blocking, interference, etc.

Additive White Gaussian Noise (AWGN) is a basic noise model used in information theory to mimic the effect of many random processes that occur in nature, which is the statistically random radio noise characterized by a wide frequency range with regards to a signal in the communications channel.

. For encoded codeword , 10-bit noise is added which corrupts the codeword. Encoded codeword is

C= [1 0 0 1 0 1 0 0 1 0 0 0 0 1 0 0] and

8-bit noise is [1 0 0 0 0 0 0 0]

[1 0 0 1 0 1 0 0 0 1 1 0 1 1 0 0] + [1 0 0 0 0 0 0 0] =

[0 0 0 1 0 1 0 0 1 0 0 0 0 1 0 0]

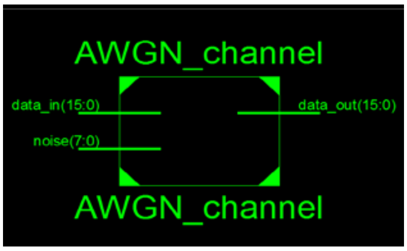


Fig 3.2 RTL schematic of AWGN channel

3.3 Decoder Implementation

At the decoder initially, received codeword is tested whether it is valid or not. Thus detector code is written in Verilog HDL which is used to detect and correct the error which is added in AWGN channel. RTL schematic for detector block obtained from Xilinx as shown in figure 3.3.1

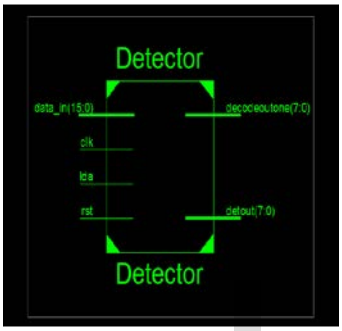


Fig 3.3.1 RTL schematic of detector

Now after correction of error decoding of the resultant code word should be done. Decoding for this decoder part is done on verilog. RTL schematic for decoder block obtained from Xilinx as shown in figure 3.3.2

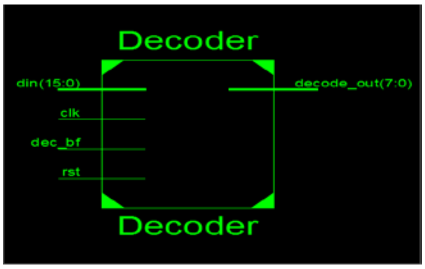


Fig 3.3.2 RTL schematic of decoder

4. Conclusion

Hence we have implemented the convolution encoder and the viterbi decoder with constraint length of K=5 and the code rate of 1/2 using Verilog HDL and simulated using the Xilinx 13.1 in Artix-7.Viterbi decoder can perform efficiently when the correcting and detecting bit is only 1,when the bits increases the efficiency decreases.Viterbi decoder have many applications such as in speech recognition,radio communication etc.

References

[1] C. E. Shannon, “A mathematical theory of communication,” Bell Sys. Tech. J., vol. 27, pp. 379–423 and 623–656, 1948.

[2] R. W. Hamming, “Error detecting and correcting codes,” Bell Sys. Tech. J., vol. 29, pp. 147–160, 1950.

[3] 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.

[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.