

# Design of High Speed Low Power Viterbi Decoder to Improve The Performance using Trace Back Architecture

Unnimaya.A.P

Student(M.Tech)

Department of Electronics and Communication,  
T.John Institute of Technology,  
Bangalore, Karnataka, India,

Narayana Swamy.R

Associate Professor

Department of Electronics and Communication,  
T.John Institute of Technology,  
Bangalore, India

**Abstract**—this paper deals with high speed, low power viterbi encoder and decoder architecture designs. Convolution (Viterbi) encoder uses the trellis diagram for the encoding and the viterbi decoder is designed to decode the encoded data. The main aim of this paper to understand the viterbi algorithm and design the encoder and decoder with constraint length 7 and code rate  $\frac{1}{2}$ .

**Keywords**—Convolution encoder, Trellis diagram, Viterbi algorithm, Viterbi decoder, Verilog HDL

## I. INTRODUCTION

The Viterbi algorithm, proposed in 1967 by Viterbi, is an efficient decoding algorithm for decoding the convolution codes and it is widely used in communication systems. The reliability and efficiency of data transmission is one of the concerning issue for communication channels. In communication systems error correction technique plays an important role. Viterbi decoding has a fixed decoding time and it is well suited for the implementation of hardware decoder. Viterbi algorithm can reduce the decoding speed and also the overall area can be increased. By using this decoding technique the overall power can be reduced and also the number of computation should be reduced. Convolutional codes are non blocking codes that can be designed to either error detecting or correcting. Convolution coding has been used in communication systems including wireless communication and deep space communication. At the receiver end using Viterbi decoder the original sequence is obtained from the received data. It implements Viterbi algorithm which is a dynamic programming algorithm, based on the minimum cumulative hamming distance it decides the optimal trellis path that is most likely followed at the encoder.

## II. CONVOLUTION ENCODER

Convolutional codes encoding can be accomplished using simple registers. The fundamental working principal of a convolution encoder is that the encoder performs a convolution of the input stream with encoder's impulse response. In convolutional encoder, the message sequence continuously runs through the encoder unlike in the block coding schemes where the message is first divided into long blocks and then encoded. Convolutional coding correlates

information elements by means of exclusive-or(XOR) operation, resulting in the increases of transmission redundancy.

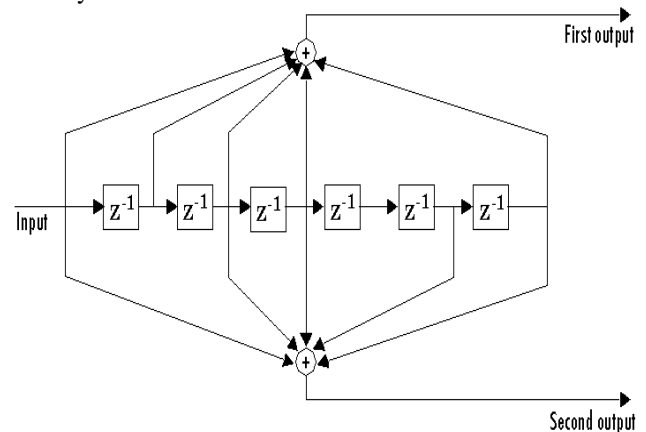


Figure 1. Convolution encoder for constraint length(K)=7, bit rate(r)=1/2

The block diagram of a convolution encoder is shown in Figure 1. To generate the output, the encoder uses 7 values of the input signal. The set of values of input data in the shift register is called the constraint length. Each set of outputs is generated by XOR ing a pattern of current and shifted values of input data. In this paper for a convolutional encoder, the following notations are used.

**c** = number of output bits.

**x** = number of input bits entering at a time.

**m** = number of stages of shift register.

**L** = number of bits in a message sequence.

**Constraint Length: K** =(m+1) digits.

**Bit rate: r** = x/c

Graphically there are three ways to understand operation of the encoder. These are state diagram, tree diagram and trellis diagram.

### A. State Diagram

Operation of a convolution encoder can be described by a state diagram. The state of the encoder is defined as the

content of its shift register. Each 'x' new bit input results in a new state. For one bit entering the encoder there are  $2_1 = 2$  possible branches for every state. If the Constraint length  $k=7$ , then size of the shift register would be  $m=6$  which results in  $2m$  states. Therefore  $2_6 = 64$  states are named from  $S_0$  to  $S_{63}$ .

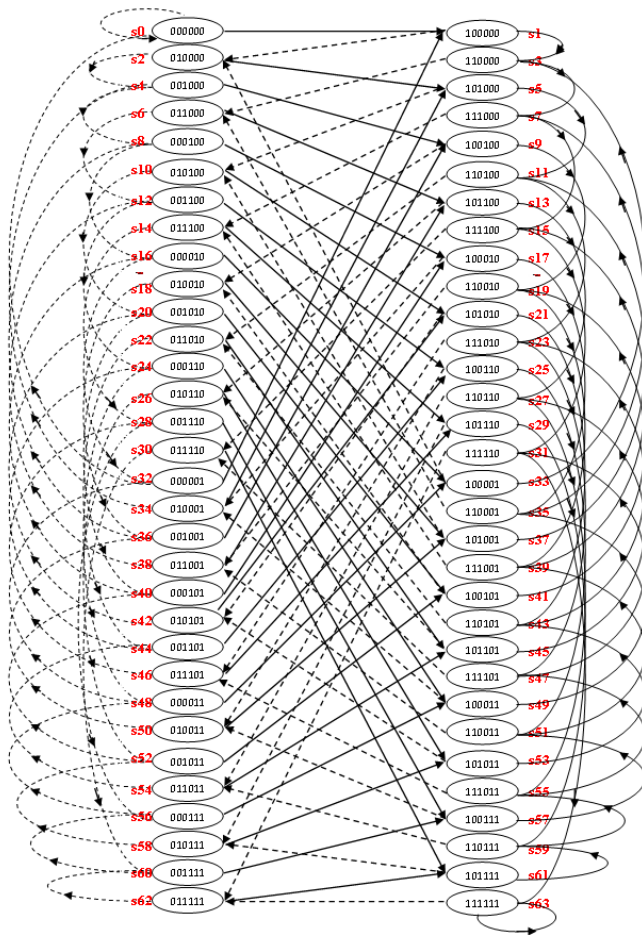


Figure 2. State diagram of  $K=7, r=1/2$  convolution encoder

State diagram is shown in figure 2. Here all the 64 states are represented and labeled as  $S_0, S_1, \dots, S_{63}$ . Consider the state  $S_0$  (000000) with '0' input, the shift register remains at same state  $S_0$  and is shown as a dotted loop starting from  $S_0$  and ending at  $S_0$ . With '1' input, the shift register moves to state  $S_1$  (100000) and is shown as a solid line starting from  $S_0$  and ending at  $S_1$ . To make easy for tracking the transition two different types of lines are used. Solid line represents the transition when the input bit is '1' and dotted line represents the transition when the input bit is '0'.

**B. Tree Diagram**

In the state diagram shown in Figure 2. It is very difficult to follow the paths because too many paths leads to confusion hence for better understanding state diagram can be re-drawn as 'code-tree'. It is better than a state diagram but still no preferred approach for representing convolutional code. If the input is a '0', then the upper path is followed and if the input is a '1', then lower path is followed, the circles represent the 'node' and lines represent the 'branch'. The output code  $[C_{(1)} C_{(2)}]$  for each input is shown on the branches.

Consider the input sequence 10110110 as the input to the convolution encoder, and then the code tree for the above input is as shown in Figure 3.

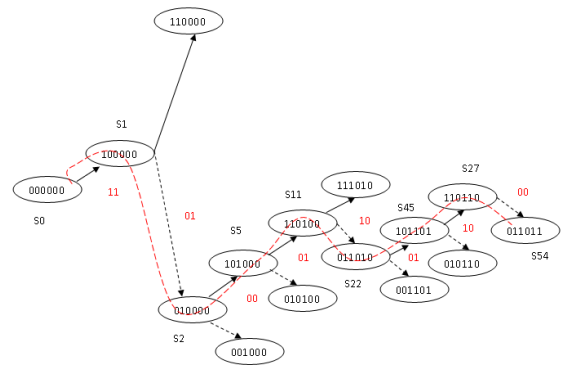


Figure 3. Code Tree for the input 10110110

**C. Trellis Diagram**

Trellis diagrams are little complex than state and tree diagram still they are more preferable for the higher constraint length. 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 trellis is a graph whose nodes are ordered into vertical slices (time), and with each node at each time connected to at least one node at an earlier and at least one node at a later time. The earliest and latest times in the trellis have only one node. Figure 4 shows the trellis diagram representation of encoding process with constraint length 7.

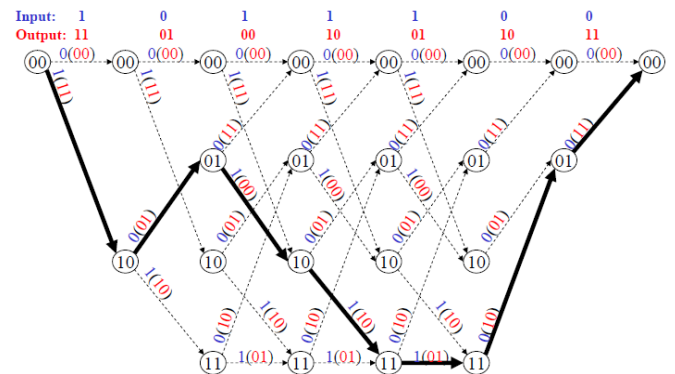


Figure 4. Trellis Diagram representation of encoding process

**III. VITERBI DECODER**

There are two categories to decoding convolutional codes. These are maximum likelihood decoding and sequential decoding based on Viterbi algorithm and Fano algorithm respectively. 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 5 shows the block diagram of Viterbi decoder.

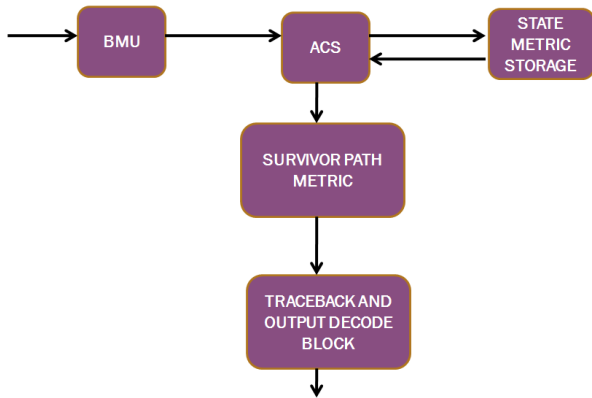


Figure 5. Block diagram of Viterbi Decoder

The basic units of viterbi decoder are

- a). Branch Metric Unit (BMU).
- b). Add Compare and Select Unit (ACS).
- c). Survivor Memory Management Unit.
- d). Trace Back Unit (TBU).

A. Branch Metric Unit (BMU)

In this unit hamming distance computation is done. This unit compares the received codes with expected codes of the current state and calculates the hamming distance between them.

The Branch Metric (BM) unit is used to calculate branch metric for all trellis branches from the input data. It is transition metric unit and its function is to generate corresponding merit of skip branch according to the input code sequence. The generated sequence merit is delivered to Add Compare Select (ACS) unit, completing the process as bit calculation. Then Branch Metric (BMs) are fed into the ACS unit.

B. Add Compare Select Unit (ACS)

Add Compare Select Unit recursively computes path metrics and outputs decision bits for each state transition. The Add Compare Select module not only receives the code sequence from the BM module, but needs the path merit of last state and information related to state shift. It is necessary to calculate the sum of branch merit and path merit firstly, and then select the smallest path merit. For a given code with rate  $1/n$  and total memory (M), the number of Add Compare Select (ACS) required to decode received sequence of length L is  $L \times 2M$ .

An Add Compare Select (ACS) module is shown in Figure 6. The two adders compute the partial path metric of each branch, the comparator compares the two partial metrics, and the selector selects an appropriate branch.

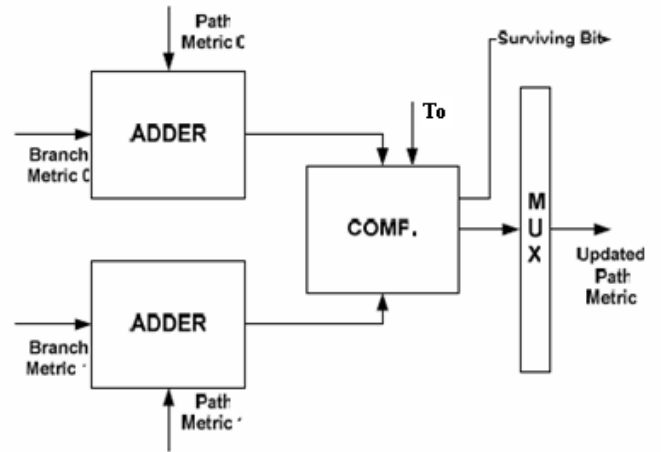


Figure 6. Add Compare Select Unit (ACS)

C. Survivor Memory Unit

Survivor Memory Unit is used for storing the survivor path values of the Add Compare Select (ACS) modules. Each stage there are 64 survivor paths and number of such stages vary depending on the length of encoded bits received. Another memory is reserved for trace back depth which defines the maximum number of stages that is allowed during the decoding process.

D. Trace Back Unit

Once the minimum path metrics (PM) of all the nodes at each stage is calculated, the minimum path metric (PM) at the last stage is found. The node having the minimum path metrics at the last stage is given as input to Trace Back Unit (TBU) and then it starts trace backing the survival paths from that node and outputs the corresponding bit which has caused the transition of that path. Figure 7 shows the Trace back procedure of optimal path.

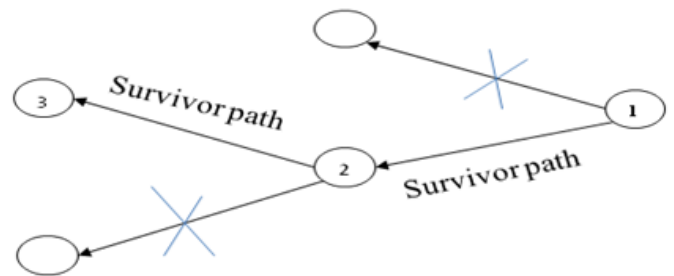


Figure 7. Trace back procedure of optimal path.

E. Trellis Diagram

The original message sequence is encoded following a path which has to be determined at the decoder part of the receiver to get the original message sequence from the encoded data. For the representation of this path a 'trellis diagram' is used. Figure 8 shows the viterbi decoding process.

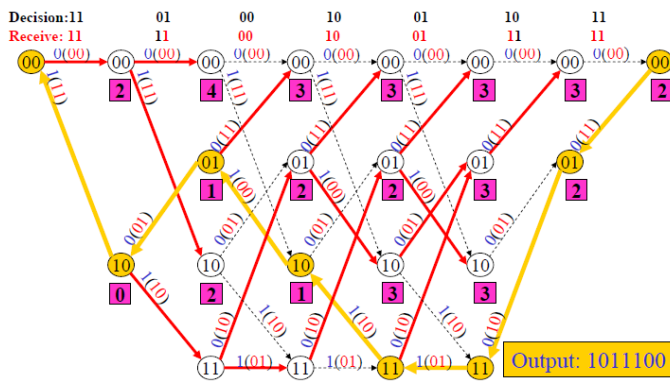


Figure 8. Viterbi Decoding Process

F. Viterbi Algorithm

In 1967 A.J Viterbi proposed an algorithm as an ‘asymptotically optimum’ approach to the decoding of convolutional codes. Viterbi decoding is based on the maximum likelihood (ML) decoding algorithm. Maximum likelihood decoding means finding the code branch in the code trellis that was most likely to be transmitted. Maximum likelihood decoding is based on calculating the hamming distances for each forming encode word. Most likely path through the trellis will maximize the metric. Viterbi algorithm performs maximum likelihood decoding by reducing its complexity. It eliminates at least likely trellis path at each transmission stage and reduce the decoding complexity. Viterbi algorithm gets its efficiency via concentrating on survival paths of the trellis. VA is an optimum algorithm for estimating the state sequence of a finite state process.

G. Results

The Convolution encoder Viterbi Decoder for constraint length 7 and bit rate 1/2 has been developed. Figure 9 shows the output of the encoder.

Input= 1 0 0 1 0 1 0  
Output= 11011111010001

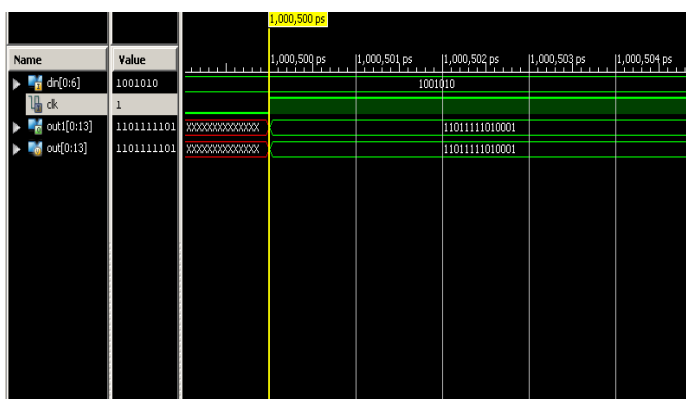


Figure 9. Output Waveform of the encoder

Considering the effects of noise suppose if the encoded data is corrupted then also decoder should be able to

retrieve the original message sequence. Figure 10 shows the output of decoder if the error has occurred in the sequence.

- a) Input= 11111111010001  
Output= 1 0 0 1 0 1 0
- b) Input= 11011011010001  
Output= 1 0 0 1 0 1 0
- c) Input=1101111110001  
Output= 1 0 0 1 0 1 0

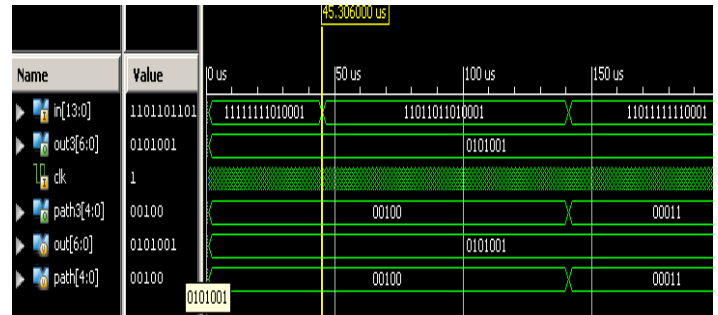


Figure 10. Output waveform the decoder

ACKNOWLEDGMENT

We take this opportunity to express our deepest gratitude and appreciation to all those who have helped us directly or indirectly towards the successful completion of this paper.

REFERENCES

- [1] Wei Chen, “RTL Implementation of Viterbi Decoder,” Master’s thesis, Linkoping Uni. Rep, pp.6-25, 2006.
- [2] Samirkumar Ranpura and Dong Sam Ha, “Low- Power Viterbi Decoder Design for Wireless Communications Applications,” Int. ASIC conference, Sept. 1999, Washington, D.C.
- [3] B. Sklar, Digital Communications: Fundamentals And Applications, Prentice-Hall, 2nd Edition, 2002.
- [4] Nirmal bhatt, Prof.Milind Shah,Prof.Bhavesh Asodariya, “Fpga Implementation of power efficiency low latency Viterbi decoder”, IJERT May-2013.
- [5] G.Feygin and P.G.Gulak, “Survivor Sequence memory management in viterbi decoders,” CSRI Tech.Rep.262, Univ.of Toronto, Jan1991.
- [6] Z. M. Patel, “VLSI implementation of IEEE802.11a Physical layer baseband”, M.Tech Dissertation, IITB, Powai. 2009.
- [7] R C S Morling and N Haron, “Novel Low-Latency Low-Power Viterbi Decoder Traceback Memory Architecture,” IEEE MELECON, Dubrovnik, Croatia, May, 2004.
- [8] S.W.Shaker, S.H.Elramly, K.A.Shehata, “FPGA Implementation of a Viterbi Decoder for WiMAX Receiver,” International Conference on Microelectronics(ICM), Marrakech, 2010.
- [9] Sang-Ho Seo, and Sin-Chong Park, “Low latency and power efficient VD using Register Exchanged state- mapping Algorithm,” Proceedings of the 9th International Database Engineering & Application Symposium (IDEAS’05).
- [10] Viterbi A. J. “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” IEEE Transactions on Information Theory13 (2): 260-269, April, 1967.