

# Design And Implementation Of Coding Techniques For Communication Systems Using Viterbi Algorithm

\* V S Lakshmi Priya<sup>1</sup>

Duggirala Ramakrishna Rao<sup>2</sup>

<sup>1</sup>PG Student (M. Tech-ECE), Dept. of ECE, Geetanjali College Of Engineering & Technology, Hyderabad, AP.

<sup>2</sup>Professor & Dean R&D, Dept. of ECE, Geetanjali College Of Engineering & Technology, Hyderabad, AP.

---

**Abstract:** Convolutional encoding with Viterbi decoding is a powerful method for error checking. It has been widely deployed in many wireless communication systems to improve the limited capacity of the communication channels. The Viterbi algorithm is the most extensively employed decoding algorithm for convolution codes. In this project, we present an implementation of Viterbi Decoder for code rate of  $\frac{1}{2}$  and for constraint length of 9 which is employed in present technologies. The Viterbi algorithm is commonly used in a wide range of communications and data storage applications. This paper discusses about architecture for a Viterbi Decoder using VLSI design techniques at circuit level. The Viterbi decoder comprises of BMU, PMU and a SMU. Communication within the decoder blocks is controlled by the Request-Acknowledge handshake pair which signals that data is ready for process. The design of various units of Viterbi Decoder is done by VERILOG HDL language. The simulation results show the decoder output of the encoded bits which are nothing but the bits which we are applied to convolutional encoder.

**Keywords:** Convolutional encoder, Viterbi decoder, Verilog HDL, Viterbi Algorithm.

---

## 1. Introduction

Coding is the important aspect of any digital communication system. Often it is done to protect the signal from getting corrected due to strong noise and interference in the channel.

Convolutional encoding with viterbi decoding is the predominant forward error correction (FEC) technique used in the most of the communication systems like Deep space communications, Wireless communications etc.,

## 2. Convolutional Encoding

A convolutional encoder typically will generate two or three output bits for each input bit. The output bits generated by the encoder are dependent on the current

input bit, as well as the state of the encoder. The state of the encoder is represented by several bits which precede the current bit. If the state of the encoder consists of the three previous bits, then there are eight possible encoder states, one for each possible combination. This encoder is said to have a constraint length  $K = 4$  since the output depends on four bits (the current bit plus three previous bits). The code rate  $r$  is defined as the number of input bits divided by the number of output bits. Thus, an encoder which produces two output bits for every input bit is said to have rate  $r = \frac{1}{2}$ .

The convolutional code in a GSM system uses a rate of  $r = \frac{1}{2}$  and  $K = 5$ , [8]

which mean that five consecutive bits are used to calculate the redundant bits and that for each data bit an additional redundant bit is added. Before the information bits are encoded, four bits are added at the end of the information bits. These bits are all set to zero and are used to reset the convolutional encoder to its initial state. Figure 2.1 shows the complete convolutional coding process for the full rate speech encoder. After the speech data passes through the parity encoder and the convolutional encoder, some of the bits have double protection, some have only a convolutional code protection, and some have no error protection. Good results were still obtained from this peculiar combination of error protection codes because the particular bits which were used for each type of error protection were carefully selected. Researchers discovered that some of the data bits produced by the full rate speech coder were much more important to perceptually good speech quality by testing many different digitally coded speech samples. The different bits have a particular role in the speech coding processing, and have different priority for various reasons. For example, any bit which comes from the most significant bit position of a number is clearly more important to the accuracy of the result than a bit from the least significant bit position.

Convolutional error correction method is used for full rate encoder. The least important Class 2 bits are not protected

at all during the convolutional coding process. The 189 bits (Class 1a with parity bits and Class 1b) enter the convolutional encoder, and  $2 \times 189 = 378$  bits emerge; the 78 Class 2 bits are appended after the 378 coded bits to yield a total of 456 bits. This is exactly 4 times 114; and 114 is the number of coded bits within one burst. This is also 8 times 57, which is the number of bits in eight sub blocks [5][6]

The following block diagram shows the working of the convolution encoder for  $r = \frac{1}{2}$ ,  $K = 5$  convolutional encoder used for speech encoding in GSM [8]. Here M1, M2, M3 and

M4 are D flip-flop registers, which are memory cells. G0 and G1 are Exclusive-OR circuits which are known as generator polynomials for Convolutional encoder for speech encoding.

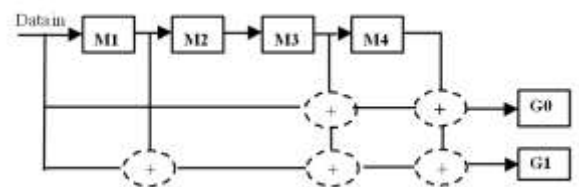


Figure 1 Convolutional Encoder

As an example, consider a 13-bit data stream in table 1.1. For a 13-bit data stream, when the coding process starts, all the memory cells M are reset to zero. The speech bits arrive sequentially at the input port. Each time a bit enters the encoder on the left; two bits come out through the two ports provided on the right of the encoder G0, G1. The table shows the state of the encoder. Row 1 is the 13 data bits. Row 2 shows the 17 bits

after appending four zero bits. Row 3 is the content of register M1 in the encoder. It is logically equivalent to the input data delayed one time period. Row 4 shows the contents of stage M2 in the encoder. It is logically equivalent to input delayed by two time periods. Similarly, M3 is delayed by 3 time periods and M5 is delayed by 4 time periods

Table 1 Example of a convolutional encoder

1	1-bit data input (13-bit)	1011000110101
2	Appending of four 0 bits	10110001101010000
3	Delay by one bit (M1)	010110001101010000
4	Delay by two bit (M2)	0010110001101010000
5	Delay by three bit (M3)	00010110001101010000
6	Delay by four bit (M4)	000010110001101010000
7	$G_0 = 1 \oplus M_1 \oplus M_2$	101011001000011111
8	$G_1 = 1 \oplus M_1 \oplus M_2 \oplus M_3$	1111010000101010111
9	Output of the convolution encoder	11011100111000001100010000111111

Row 7 shows encoder output G0; Row 8 shows encoder output G1. Row 9 shows the output pair (G1 G0) at each time period. Recall that the (r=1/2) encoder produces two output bits for each input bit. In general, the performance of a convolutional encoder will improve as the constraint length increases, or as the code rate decreases. These codes are described by generator polynomials which describe relationships between the current input and state bits, and the resulting output bits. Therefore Row 9 shows the output of the encoder, these encoded data is the input to the Viterbi decoder.

### 3. Viterbi Decoding

A viterbi decoder uses the Viterbi algorithm for decoding a bitstream that has been encoded using Forward error correction based on a Convolutional code. The Viterbi algorithm is commonly used in a wide range of communications and data storage applications. It is used for decoding convolutional codes, in baseband detection for wireless systems, and also for detection of recorded data in magnetic disk drives. The requirements for the Viterbi decoder or

Viterbi detector, which is a processor that implements the Viterbi algorithm, depend on the applications where they are used. This results in very wide range of required data throughputs and power or area requirements.

Viterbi detectors are used in cellular telephones with low data rates, of the order below 1Mb/s but with very low energy dissipation requirement. They are used for trellis code demodulation in telephone line modems, where the throughput is in the range of tens of kb/s, with restrictive limits in power dissipation and the area/cost of the chip. On the opposite end, very high speed Viterbi detectors are used in magnetic disk drive read channels, with throughputs over 600Mb/s. But at these high speeds, area and power are still limited.

Convolutional coding has been used in communication systems including deep space communications & wireless communications. It offers an alternative to block codes for transmission over a noisy channel. An advantage of convolutional coding is that it can be applied to a continuous data stream as well as to blocks of data. IS-95, a wireless digital cellular standard for CDMA (code division multiple access), employs convolutional coding. A third generation wireless cellular standard, under preparation, plans to adopt turbo coding, which stems from convolutional coding.

There are other algorithms for decoding a convolutionally encoded stream (for example, the Fano algorithm). The Viterbi algorithm is the most resource-consuming, but it does the maximum likelihood decoding. It is most often used for decoding convolution codes with constraint lengths  $k \leq 10$ , but values up to  $k=15$  are used in practice. There are both hardware (in modems) and software implementations of a Viterbi decoder.

**Architecture of proposed viterbi algorithm**

A hardware viterbi decoder for basic code usually consists of the following major blocks as shown in fig-2:

- \* Branch metric unit (BMU)
- \* Path metric unit (PMU)
- \* Trace back unit (TBU)

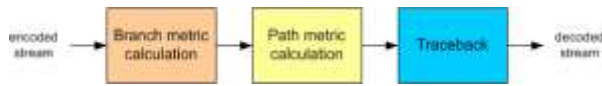


Figure 2 Viterbi Decoder

In the viterbi decoding approach the trace-back (TB) and the register-exchange (RE) methods are the two major techniques used for the path history management in the chip designs of Viterbi decoders. The TB method takes up less area but requires much more time as compared to RE method because it needs to search or trace the survivor path back sequentially. But the major disadvantage of the RE approach is that its routing cost is very high especially in the case of long-constraint lengths and it requires much more resources.

**BRANCH METRIC UNIT:-**

A branch metric unit's function is to calculate branch metrics, which are normed distances between every possible symbol in the code alphabet, and the received symbol as shown in fig-2.

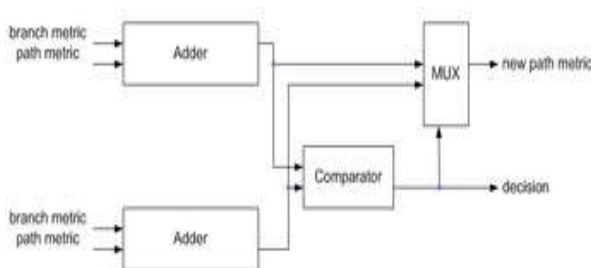


Figure 3 Decision circuit

There are hard decision and soft decision viterbi decoders. A hard decision viterbi decoder receives a simple bitstream on its input, and a Hamming distance is used as a metric. A soft decision viterbi decoder receives a bitstream containing information about the reliability of each received symbol. Usually this reliability information is encoded as follows (the table is shown for 3-bit input):

value	meaning
000	strongest 0
001	relatively strong 0
010	relatively weak 0
011	weakest 0

Of course, it is not the only way to encode reliability data. The squared Euclidean distance is used as a metric for soft decision decoders.

**Path metric unit (PMU):-**

A path metric unit summarizes branch metrics to get metrics for  $2K - 1$  paths, one of which can eventually be chosen as optimal. Every clock it makes  $2K - 1$  decisions, throwing off wittingly non-optimal paths. The results of these decisions are written to the memory of a trace back unit. Fig-4 shows the pmu.

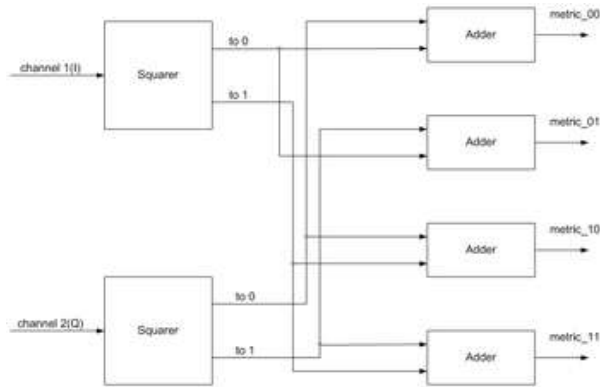


Figure 4 The PMU circuit

The core elements of a PMU are ACS (Add-Compare-Select) units. The way in which they are connected between themselves is defined by a specific code's trellis diagram. Fig-4 shows the ACS. Since branch metrics are always ones and zeros, there must be an additional circuit preventing metric counters from overflow. An alternate method that eliminates the need to monitor the path metric growth is to allow the path metrics to "roll over", to use this method it is necessary to make sure the path metric accumulators contain enough bits to prevent the "best" and "worst" values from coming within  $2(n-1)$  of each other. The compare circuit is essentially unchanged. It is possible to monitor the noise level on the incoming bit stream by monitoring the rate of growth of the "best" path metric, a simpler way to do this is to monitor a single location or "state" and watch it pass "upward" through say four discrete levels within the range of the accumulator. As it passes upward through each of these thresholds, a counter is incremented that reflects the "noise" present on the incoming signal.

**Trace back unit (TBU):-**

Back-trace unit restores an (almost) maximum-likelihood path from the decisions made by PMU. Since it does it in inverse direction, a viterbi decoder comprises a FILO (first-in-last-out) buffer to reconstruct a correct order. Fig-5 shows the TBU.

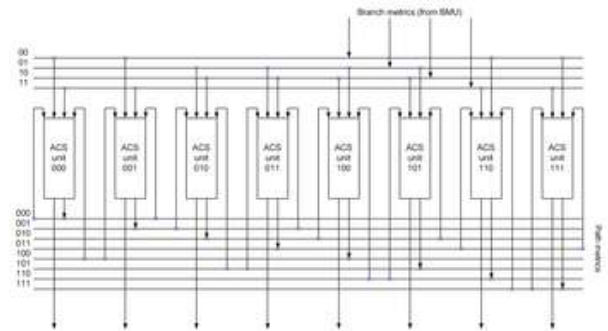


Figure 5 Trace Back Unit

Note that the implementation shown on the image requires double frequency. There are some tricks that eliminate this requirement. However, computing the node which has accumulated the largest cost (either the largest or smallest integral path metric) involves finding the maxima or minima of several (usually 2K-1) numbers, which may be time consuming when implemented on embedded hardware systems.

Most communication systems employing Viterbi decoding involving data packets of fixed sizes, with a fixed bit/byte pattern either at the beginning or/and at the end of the data packet. By using the known bit/byte pattern as reference, the start node may be set to a fixed value, thereby obtaining a perfect Maximum Likelihood Path during trace back.



### Trellis Diagram

A convolutional encoder is often seen as a finite state machine. Each state corresponds to some value of the encoder's register. Given the input bit value, from a certain state the encoder can move to two other states. These state transitions constitute a diagram which is called a trellis diagram. A trellis diagram is depicted on the Figure 6. A solid line corresponds to input 0, a dotted line – to input 1 (note that encoder states are designated in such a way that the rightmost bit is the newest one).

Each path on the trellis diagram corresponds to a valid sequence from the encoder's output. Conversely, any valid sequence from the encoder's output can be represented as a path on the trellis diagram. One of the possible paths is denoted as red (as an example). Note that each state transition on the diagram corresponds to a pair of output bits. There are only two allowed transitions for every state, so there are two allowed pairs of output bits, and the two other pairs are forbidden. If an error occurs, it is very likely that the receiver will get a set of forbidden pairs, which don't constitute a path on the trellis diagram. So, the task of the decoder is to find a path on the trellis diagram which is the closest match to the received sequence.

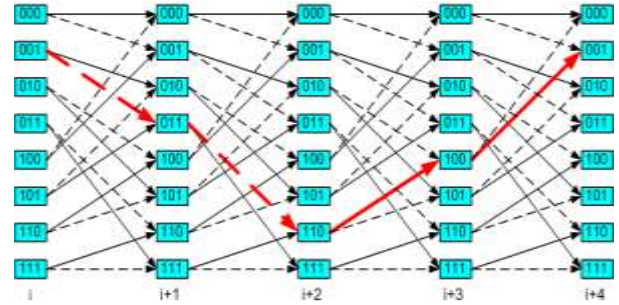


Figure 6 A trellis diagram corresponding to the encoder on the Figure 2

Let's define a free distance  $d_f$  as a minimal Hamming distance between two different allowed binary sequences (a Hamming distance is defined as a number of differing bits).

### 4. Applications

The Viterbi decoding algorithm is widely used in the following areas:

- \* Decoding trellis-coded modulation (TCM), the technique used in telephone-line modems to squeeze high spectral efficiency out of 3 kHz-bandwidth analog telephone lines. The TCM is also used in the PSK31 digital mode for amateur radio and sometimes in the radio relay and satellite communications.
- \* Decoding convolutional codes in satellite communications.
- \* Computer storage devices such as hard disk drives.
- \* Hidden Markov model (HMM)-based speech recognition

A viterbi decoder uses the Viterbi algorithm for decoding a bitstream that has been encoded using Forward error correction based on a Convolutional code. The Viterbi algorithm is commonly used in a wide range of communications and data storage applications. It is used for decoding

convolutional codes, in baseband detection for wireless systems, and also for detection of recorded data in magnetic disk drives. The requirements for the Viterbi decoder or Viterbi detector, which is a processor that implements the Viterbi algorithm, depend on the applications where they are used. This results in very wide range of required data throughputs and power or area requirements. Viterbi detectors are used in cellular telephones with low data rates, of the order below 1Mb/s but with very low energy dissipation requirement. They are used for trellis code demodulation in telephone line modems, where the throughput is in the range of tens of kb/s, with restrictive limits in power dissipation and the area/cost of the chip. On the opposite end, very high speed Viterbi detectors are used in magnetic disk drive read channels, with throughputs over 600Mb/s. But at these high speeds, area and power are still limited.

Convolutional coding has been used in communication systems including deep space communications and wireless communications. It offers an alternative to block codes for transmission over a noisy channel. An advantage of convolutional coding is that it can be applied to a continuous data stream as well as to blocks of data. IS-95, a wireless digital cellular standard for CDMA (code division multiple access), employs convolutional coding.

#### **4. Results & Conclusions**

In this paper, a high-speed VD design for lossless systems. The computation architecture that incorporates algorithm

efficiently improves the VDs speed without making any penalty on the area parameter appreciably. We have also analyzed the precomputation algorithm, where the optimal precomputation steps are calculated and discussed. This algorithm is suitable for TCM systems which always employ convolutional codes. Finally, we presented a design case. Both the ACSU and SMU are modified to correctly decode the signal. Synthesis and simulation estimation results show that there is no loss of information while transmitting the binary information

The Convolutional encoder for the constraint length of  $K=9$  and code rate of  $r=1/2$  has been developed and the synthesis is carried out. It has been simulated and the simulation result is shown in fig. 7. The Viterbi decoder has been developed using Xilinx ISE 8.2i and the synthesis is carried out.

It (decoder) has been simulated and the simulation result is shown in fig. 8.

Typical input and output are as indicated below.

Input bits  $X=[010111001010001]$

Output bits  $Y=[00\ 11\ 10\ 00\ 01\ 10\ 01\ 11\ 11\ 10\ 00\ 10\ 11\ 00\ 11]$

Encoded noise  $= [00\ 11\ 11\ 00\ 01\ 10\ 01\ 11\ 11\ 10\ 10\ 10\ 11\ 00\ 11]$

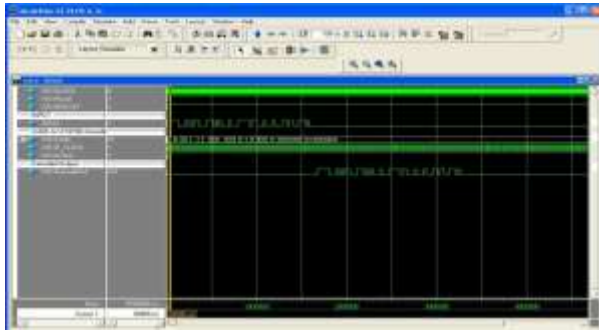


Figure 7 Simulation Results

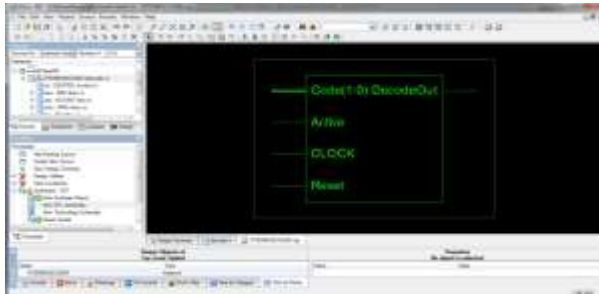


Figure 8 Schematic diagram



Figure 9 RTL Schematic

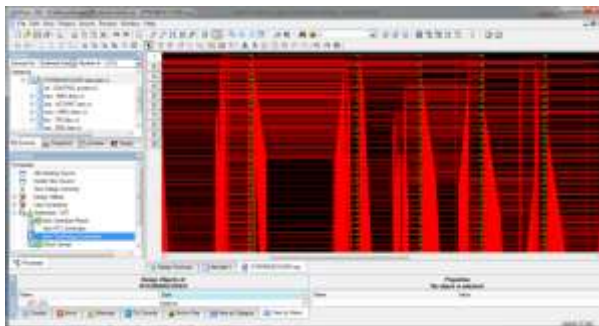


Figure 10 Technology Schematic

### Acknowledgements

The authors would like to thank the anonymous reviewers for their comments which were very helpful in improving the quality and presentation of this paper.

### References:

- [1] R. A. Abdallah and N. R. Shanbhag, "Error-resilient low-power viterbi decoder architectures," *IEEE Trans. Signal Process.*, vol. 57, no. 12, pp. 4906–4917, Dec. 2009.
- [2] J. B. Anderson and E. Offer, "Reduced-state sequence detection with convolutional codes," *IEEE Trans. Inf. Theory*, vol. 40, no. 3, pp. 965–972, May 1994.
- [3] C. F. Lin and J. B. Anderson, "Viterbi algorithm decoding of channel convolutional codes," presented at the Princeton Conf. Info. Sci. Syst., Princeton, NJ, Mar. 1986.
- [4] S. J. Simmons, "Breadth-first trellis decoding with adaptive effort," *IEEE Trans. Commun.*, vol. 38, no. 1, pp. 3–12, Jan. 1990.
- [5] F. Chan and D. Haccoun, "Adaptive viterbi decoding of convolutional codes over memoryless channels," *IEEE Trans. Commun.*, vol. 45, no. 11, pp. 1389–1400, Nov. 1997.
- [6] "Bandwidth-efficient modulations," Consultative Committee For Space Data System, Matera, Italy, CCSDS 401(3.3.6) Green Book, Issue 1, Apr. 2003.
- [7] J. Jin and C.-Y. Tsui, "Low-power limited-search parallel state viterbi decoder implementation based on scarce state transition," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 11, pp. 1172–1176, Oct. 2007.
- [8] F. Sun and T. Zhang, "Lowpower state-parallel relaxed adaptive viterbi decoder design and implementation," in *Proc. IEEE ISCAS*, May 2006, pp. 4811–4814.



**Authors Profile:**



**V S LAKSHMI PRIYA** is Pursuing M. Tech From Geetanjali College Of Enginnering & Technology, JNTUH with specialization in Electronics & Communications Engineering (ECE). She completed her B.Tech from Vignana Bharathi Institute Of Technology, JNTUH with specialization in ECE in the year 2010.



**Duggirala Ramakrishna Rao** graduated in Engineering (B.E. in ECE) from Government College of Engineering (Presently College of Engineering ,JNTUK), Kakinada, completed Post Graduation (M.Tech in Electronic Instrumentation) from Regional Engineering College(Presently NIT Warangal),Warangal. He has vast experience in the fields of Image Processing, Remote Sensing Radars and LIDARs. He has published and presented about 40 technical papers at the national and international level journals and conferences. He worked for the prestigious Indian Space Research Organisation (ISRO) about 35 years and was the Deputy Project Director of Space Borne Lidar Project in ISRO. Presently he is the Professor in ECE department and Dean R & D of Geetanjali College of engineering and Technology. He is the Fellow of IETE and senior member of ISTE.