

# Design and Implementation of Efficient Viterbi Decoders

K.S. Arunlal and S.A. Hariprasad

R.V. Center for Cognitive Technologies, Bangalore, Karnataka, India  
arunlalks1@yahoo.com, harivat2002@yahoo.co.in

**Abstract.** Viterbi decoders are used for forward error correction, but the algorithm demands more hardware, memory and computational time, hence researchers have come up with other alternatives like fangled viterbi decoder, modified fangled viterbi decoder, but these methods lack error correction capabilities. In this work an innovative method is used to improve error correction capabilities. The results shows it can correct two bit error with less computational time and hardware requirement.

**Keywords:** Constraint length, hamming distance, path metric, branch metric, Trellis diagram.

## 1 Introduction

In the present scenarios, data transferring between the systems plays a vital role as the technologies are increasing day-by-day, the number of users is simultaneously increasing. This wide usage leads to major issues in the digital communication systems and results in data corruptions. It's very much essential to have a technique to correct these errors efficiently. One such algorithm used by telecommunication industry is Viterbi algorithm. The major issues in Viterbi decoder is high computational complexity, large memory requirement.

Fangled viterbi decoder is a variant of viterbi decoders which has least computational complexity, delay and memory requirement because it calculates only one path in the trellis diagram which affects error detection capability.

Modified fangled takes it a step further by gaining one bit error correction and detection capability, with increased computational complexity and memory requirement compared with fangled Viterbi decoder but it is lower than Viterbi decoder.

But in reality all parameters such as error correction capability, memory requirement, design complexity and cost are interconnected with each other. In order to solve these problems, an efficient fangled viterbi algorithm is designed in this paper with 2 bit error correction capability with less hardware complexity and computational time.

The paper is organized as follows. Section 2 describes Viterbi decoding algorithm, section 3 describes fangled viterbi decoding algorithm of convolutional codes, section 4 gives description of modified fangled viterbi algorithm, section 5 describes efficient fangled viterbi decoding algorithm and section 6 concludes this paper with results.

## 2 Viterbi Decoding Algorithm

The Viterbi decoding algorithm proposed by A. J. Viterbi [1] performs maximum likelihood sequence detection on data which have been convolutionally coded. The decoding uses the trellis diagram to determine the output data sequence path which is the most likely to the input sequences from encoder. The basic units of Viterbi decoder (VD) [2] are branch metric unit (BMU), add compare and select unit (ACS) and survivor memory management unit (SMU) as shown in figure 1.

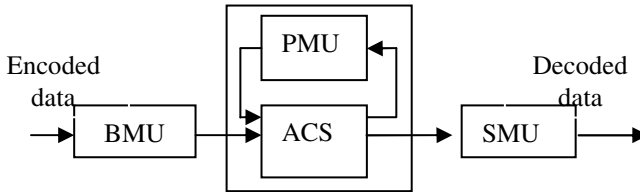


Fig. 1. Viterbi Decoder Block Diagram

Branch metric unit (BMU) compares the received data symbols to the ideal outputs of the encoder from the transmitter and hence calculate the branch metrics. Path metric computation unit (PMU) calculates the path metrics of a stage by adding the branch metrics associated with a received symbol to the path metrics from the previous stage of the trellis. An ACS module adds each incoming branch metric of the state to the corresponding path metric and compares the two results to select a smaller one. Then it will update the path metric storage with selected value. To keep track of information bits associated with the surviving paths, survivor memory management is used.

## 3 Fangled Viterbi Decoding Algorithm

It's almost similar to Viterbi decoder and it consists following functional units[3].

- Encoder Engine
- Branch metric generator
- Add-Compare-Select unit (ACSU)

The encoder Engine mimics the convolutional encoder and calculates next state and output for the various given states as shown in table 1. The branch metric unit calculates the branch metrics by using hard decision based on hamming distance as it is easy to implement. Hence it takes less hardware, computational time with no error correction capabilities.

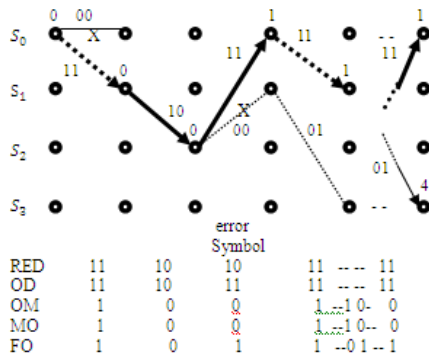
**Table 1.** State and Output table for Viterbi Decoder

Start State	Input Symbol	End State	Output Bit
00	00	00	0
00	11	01	1
01	10	10	0
01	01	01	1
10	11	00	1
10	00	01	0
11	01	10	1
11	10	11	0

The working principle of Fangled Viterbi decoder can be explained with reference to the trellis diagram, for the paths corresponding to input sequence [11 10 10 11 01 01 11]. One bit error is introduced at sixth bit position or third symbol of the original sequence as shown in figure 2. The normal line is for output data bit ‘0’ and dashed line is for ‘1’ the constraint length is taken as (K) =3 encoder, giving rise to four states in Trellis diagram. Dark line is used for correct path and ‘X’ is for wrong branch like branch 00 on the first frame. The abbreviations which are used in the Trellis diagram are:

- Received erroneous data (RED).
- Original data from encoder (OD).
- Original message to encoder (OM).
- Modified fangled decoder output (MO).
- Fangled decoder output (FO).

As shown in Figure 2 no identifiable path is there for conflicts on third symbol as there is no guarantee that state transition will be from state 2 to state 0 (correct) or state 1 (incorrect) i.e. 10 to 00 (correct) or 11. So in case of an error when selector unit in the ACS finds itself in a dilemma of choosing among equal paths, makes an arbitrary branch selection (here branch 00) leading to serious unreliability of the data thus decoded.



**Fig. 2.** Trellis diagram of path followed by sequence [11 10 10 11 01 01 11] from K=3, R=1/2, (2, 1, 2) encoder

Advantage of fangled decoder is, for every symbol only one path is stored and two additions and one comparison are done. So for 14 bit received data i.e. 7 symbols only 14 add and 7 comparisons are done [4] and corresponding memory requirement for storing path and state metric, states and survivor as well as decoded data is 88 bits [6].

#### 4 Modified Fangled Viterbi Algorithm

Modified Fangled Viterbi algorithm is a variant of fangled Viterbi algorithm designed to correct one bit errors [5]. Computational complexity and memory requirement is more than fangled but much lower than conventional Viterbi decoder. The main difference is that modified fangled takes both paths in case of a conflict or error and compares them at the end for the choice of optimum path [6]. It fails in case of noisy channel which introduces burst error. The rest of the functional blocks is similar to fangled Viterbi decoder.

If an error occurs in the received data as in third symbol shown in Figure 2, then the branch metric unit of that error frame has two possible states 00 (correct) and 10 with equal metric, leading to a conflict of branch selection between 11 (correct) and 00. In fangled algorithm this conflicting situation is resolved by randomly selecting any path which leads to erroneous decoding of data [4]. But this miscalculation is solved in modified fangled algorithm by selecting both paths and calculating their total path metric from initial state to completion of trellis and storing the error metrics of each path which came as 1 and 4. Both paths are now compared and the minimum path with metric 1 is selected, consequently resolving the issue and giving error free data [1 0 0 1 1]. Note that for error free data both paths will coincide till the end.

Advantage of modified fangled algorithm is its 1 bit error correcting capability giving reliability to the decoder. But it carries the calculation of both the path metrics from the initial stage with and without error, even irrespective of the stage at which the conflict of branch selection or error occurs. Hence leading to 28 adds and 14 compares for 14 bit received data from K=3 encoder. The second path requires additional memory, enhancing memory requirement to 256 bits [6].

#### 5 Efficient Fangled Viterbi Algorithm

Efficient fangled viterbi decoder is an extension of fangled and similar to modified fangled but its decision unit is highly efficient. This removes the redundancies in modified fangled design like extra computations and consequent memory requirement.

The whole design can again be bifurcated into:

- Encoder engine
- Branch metric calculation unit (BMU)
- Decision unit
- Survivor path calculation unit (SMU)

Decision unit works same as ACSU in fangled Viterbi decoder on uncorrupted data but in case of errors, it is more efficient for finding correct path. This work assume that one

bit error can cause the path metric to increase by one. When such happens modified fangled used to calculate both path and compare them at the end [6]. Efficient fangled method does so by taking future symbol 11 from received data i.e. the very next two bits after error symbol to compute path metric for choosing correct path.

In Figure 2, one bit error occurred at third symbol i.e. ‘10’ increasing the path metric from 0 to 1. Now fourth symbol i.e. ‘11’ is taken and path metric then calculated increases to 2. This leads to the conclusion that this path is wrong, since correct path metric always increases to one for one bit error and remains the same thereafter. Hence further calculation of this path is terminated. In case of modified fangled both paths coincide.

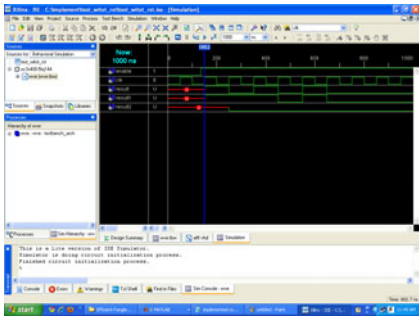


Fig. 3. Simulation result of Efficient Fangled Viterbi decoder for 1bit error at symbol 2

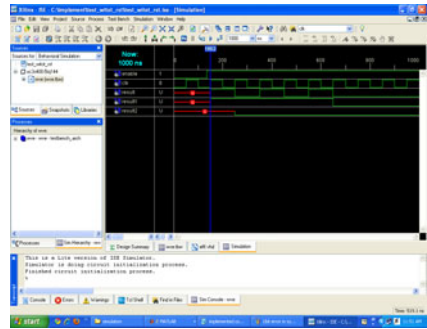


Fig. 4. Simulation result of Efficient Fangled Viterbi decoder for 2 bit error at symbol 2 & 4

Table 2. Comparison Table for decoding 14 bit input data from (2, 1, 2) encoder of constraint length 3

Parameters	Fangled VD [2]	Modified Fangled VD [2]	Efficient Fangled VD	Efficient Fangled VD	Conventional VD [2]-[3]
Error correction capability (Bits)	0	1	1	2	2
Processing time	150 ns	300 ns	150 ns	150 ns	650 ns
Number of additions required	14	28	16-18	18-22	46
Number of comparisons required	7	14	8-9	9-11	20
Device Utilization (Gate counts)	1019	1194	1542	1758	5242

First and foremost advantage is the reduction of computational complexity to 36-43% of Modified fangled Viterbi decoder for one bit error in 14 bit length data and 64%-78% for 2 bit error in 14 bit data. As for 14 bit data i.e. 7 symbols modified fangled Viterbi decoder takes 28 additions and 14 comparisons while efficient fangled can take either 2 or 4 additions and 1 or 2 comparisons respectively for various error positions as explained earlier. When no error is present this decoder will work as Fangled Viterbi decoder taking only 14 additions and 7 comparisons to decode 14 bit input data.

## 6 Results and Conclusion

Figure 3 and 4 shows the result of efficient Viterbi decoder obtained for one and two bit errors. The table 2 compares the result obtained for different variants of Viterbi decoders. From the table it is very clear that efficient fangled Viterbi decoder can perform well and also, can correct two bit errors when been compared with other viterbi variants, but with slight increase hardware complexity.

## References

1. Viterbi, A.J.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inform. Theory* IT-13, 260–269 (1967)
2. Hema, S., Suresh Babu, V., Ramesh, P.: FPGA Implementation of Viterbi Decoder. In: *Proceedings of the 6th WSEAS Int. Conf. on Electronics, Hardware, Wireless and Optical Communications*, Corfu Island, Greece, February 16-19 (2007)
3. Shivasubramaniam, R., Varadhan, A.: An efficient implementation of IS-95A CDMA Transceiver through FPGA. *DSP Journal* 6(1) (September 2006)
4. Ur Rahman, S.U., Kulkarni, S., Sujatha, C.M.: An alternative scheme for Viterbi decoding of convolutional codes. In: *NLPPES 2010*, Bangalore (2010)
5. Kulkarni, S.: Implementation of convolution- nal Viterbi decoder on FPGA and measuring its performance. MTEch thesis in R.V.C.E., Bangalore (2009-2010)
6. Ur Rahman, S.U.: Implementation of fangled Viterbi decoder on FPGA and measuring its performance. MTEch thesis in R.V.C.E., Bangalore (2009-2010)