Implementation of Viterbi Decoder on FPGA to Improve Design

Palak Gohel[1] Prof. K. C. Dave[2]
[1, 2]Instrumentation & Control Engineering Department,
[1, 2] L. D. College of Engineering, Gujarat Technological University, Ahmedabad-380015

Abstract—In the data transmissions over wireless channels are affect by attenuation, distortion, interference and noise, which affects the receiver's ability to receive correct information. Convolution coding with Viterbi decoding is a FEC technique that is particularly suited to a channel in which transmitted signal is corrupted mainly by additive white Gaussian noise (AWGN).Convolutional codes are used for error correction. They have rather good correcting capability and perform well even on very bad channels with error probabilities. Viterbi decoding is the best technique for decoding the Convolutional codes but it is limited to smaller constraint lengths. Viterbi algorithm is a well-known maximum-likelihood algorithm for decoding of convolutional codes.

Keywords: Convolution encoder, Constraint length, Viterbi decoder, FPGA, Traceback Methods

I. INTRODUCTION

Convolutional encoding is a forward error correction technique that is used for correction of errors at the receiver end. Forward error correction (FEC) much simpler techniques [1]. It is known that noise-immunity is attributes of information transmission system. Since errors are possible in communication channels during data transmission we must apply error correction codes to combat these errors. The purpose of forward error correction (FEC) is to improve the capacity of channel by adding some carefully designed redundant information to the data being transmitted through the channel. The process of adding this redundant information is known as channel coding[2]. The encoder adds redundant bits to the sender's bit stream to create a code word. The decoder uses the redundant bits to detect and/or correct as many bit errors as the particular error control code will allow. Like any error correcting code, a Convolutional code works by adding some structured redundant information to the user's data and then correcting errors using this information. There have been a few Convolutional decoding methods such as sequential and Viterbi decoding, of which the most commonly employed technique is the Viterbi Algorithm (VA).

In 1967, Viterbi developed the Viterbi Algorithm (VA) as a method to decode convolutional codes. Then, many researchers have expanded on Viterbi's work by to improve convolutional codes, exploring the performance limits of the technique, and varying decoder design parameters to optimize the implementation of the technique in hardware and software [5].

The Viterbi algorithm essentially perform maximum Likelihood decoding to correct the errors in received data which are caused by the channel noise; however it reduces the computational load by taking advantage of special structure in the code trellis. The Viterbi algorithm (VA) is a recursive optimal solution to the problem of estimating the state sequence of a discrete time finite state Markov process [6].

II. VITERBI DECODER

Overview of the basic Viterbi decoding system is illustrated in Fig. 1.

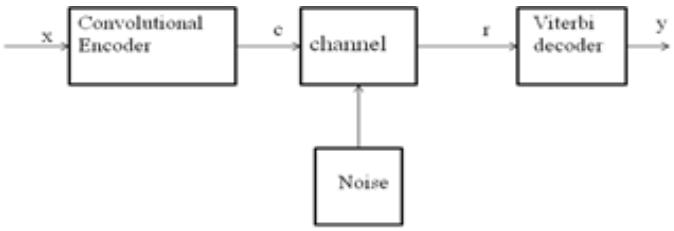

Fig. 1: Simple Viterbi decoder system

This figure shows three basic elements of the Viterbi decoding system: convolutional encoder, communication channel and Viterbi decoder.

A. Convolutional Encoder

Convolution codes were first introduced by Elias in 1955 as an alternative to block codes. Convolution codes differ from block codes in that the encoder contains memory and the n encoder outputs at any given time unit depend not only on the k inputs at that time unit but also on m previous input blocks. An (n, k, m) convolutional code can be implemented with a k-input, n-output linear sequential circuit with input memory m. typically, n and k are small integers with k < n, but the memory order m must be made large to achieve low error probabilities.

The constraint length K of the code represents the number of bits in the encoder memory that effect the generation of the n output bits and is defined as K = m+1. The code rate r of the code is a measure of the code efficiency and is defines as r= k/n. A convolutional Encoder is a Finite state machine i.e. a model of behaviour composed of states, action and transition [7]. Contents of first K-1 shift register stages defines the encoder state. Memory register start with 0 and modulo-2 adders among the registers and input generate the encoded data. Generator polynomial defines how the adders (XOR gates) are placed.

B. Viterbi Algorithm

Viterbi algorithm was introduced in 1967 by Viterbi. Viterbi algorithm is called as optimum algorithm because it minimizes the probability of error. The algorithm can be broken down into the following three steps

1) Weigh the trellis; that is, calculate the branch metrics

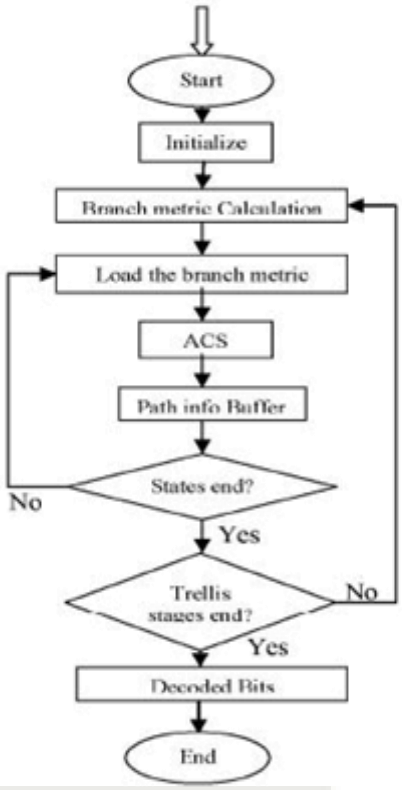

Fig. 2: Flow Chart of Viterbi Algorithm

2) Recursively computes the shortest paths to time n, in terms of the shortest paths to time n-1 In this step, decisions are used to recursively update the survivor path of the signal. This is known as add compare-select (ACS) recursion.
3) Recursively finds the shortest path leading to each trellis state using the decisions from Step 2. The shortest path is called the survivor path for that state and the process is referred to as survivor path decode. Finally, if all survivor paths are traced back in time, they merge into a unique path, which is the most likely signal path [8].

C. Viterbi Decoder

The basic units of Viterbi decoder are branch metric unit, add compare and select unit and survivor memory management unit [10].

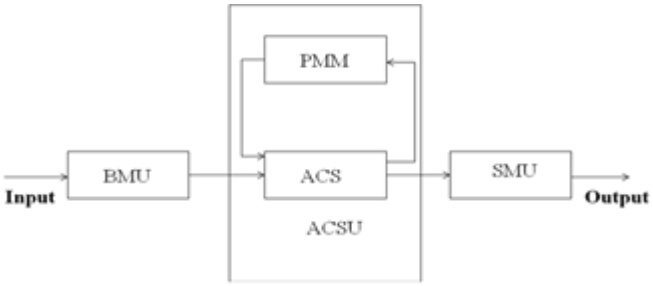

Fig. 3: Block Diagram of Viterbi decoder

1) Branch Metric Unit

The first unit is called branch metric unit. Here the received data symbols are compared to the ideal outputs of the encoder from the transmitter and branch metric is calculated. Hamming distance or the Euclidean distance is used for branch metric computation.

2) Path Metric Unit

The second unit, called path metric computation unit, 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.

Survivor Memory Management Unit

The final unit is the trace-back process or register exchange method, where the survivor path and the output data are identified. 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. Also, extra hardware is required to reverse the decoded bits. 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 [9].

3) Traceback and Register Exchange method

In the TB method, the storage can be implemented as RAM and is called the path memory. Comparisons in the ACS unit and not the actual survivors are stored. After at least L branches have been processed, the trellis connections are recalled in the reverse order and the path is traced back through the trellis diagram. The TB method extracts the decoded bits; beginning from the state with the minimum PM. Beginning at this state and tracing backward in time by following the survivor path, which originally contributed to the current PM, a unique path is identified. While tracing back through the trellis, the decoded output sequence, corresponding to the traced branches, is generated in the reverse order. Trace back architecture has a limited memory bandwidth in nature, and thus limits the decoding speed.

The register exchange (RE) method is the simplest conceptually and a commonly used technique. Because of the large power consumption and large area required in VLSI implementations of the RE method, the trace back method (TB) method is the preferred method in the design of large constraint length, high performance Viterbi decoders. In the register exchange, a register assigned to each state contains information bits for the survivor path from the initial state to the current state. In fact, the register keeps the partially decoded output sequence along the path [11].

Fig. 4: Register Exchange Method

The register-exchange method eliminates the need to trace back since the register of the final state contains the decoded output sequence. However, this method results in complex hardware due to the need to copy the contents of all the registers in a stage to the next stage. The survivor path information is applied to the least significant bit of each

register, and all the registers perform a shift left operation at each stage to make room for the next bits.

III. TYPES OF VITERBI DECODER

A. Hard decision Viterbi decoding

In the hard-decision decoding, the path through the trellis is determined using the Hamming distance measure. Thus, the most optimal path through the trellis is the path with the minimum Hamming distance. The Hamming distance can be defined as a number of bits that are different between the observed symbol at the decoder and the sent symbol from the encoder. Furthermore, the hard decision decoding applies one bit quantization on the received bits.

B. Soft decision Viterbi decoding

Soft-decision decoding is applied for the maximum likelihood decoding, when the data is transmitted over the Gaussian channel, On the contrary to the hard decision decoding, the soft-decision decoding uses multi-bit quantization for the received bits, and Euclidean distance as a distance measure instead of the hamming distance. The demodulator input is now an analog waveform and is usually quantized into different levels in order to help the decoder decide more easily. A 3-bit quantization results in an 8-ary output.

IV. OTHER TECHNIQUES OF VITERBI DECODING

A. Hybrid Register Exchange Method

Drawback of RE method is its frequent switching activity and long constraint length is overcome by combining theREM and TB techniques in Hybrid REM method. The initial state can be first traced back through an m cycle, and then transfer content of initial state to current state and the next m bits of register is the m bits of current state itself.

B. Memoryless Register Exchange Method

The implementation of memory less Viterbi Decoder (MLVD) needs to trace only one row so, the MLVD requires only one pointer to trace the current position of the decoder in the trellis.

C. Memoryless Hybrid Register Exchange Method

The MLVD keeps track of the current state position of the decoder in the memory unit. It makes use of the fact that the bit appended to each row of memory is exactly the bit that is shifted into the pointer to form the new pointer to that row of memory.

A Low Power Viterbi Decoder Design With Minimum Transition Hybrid Register Exchange Processing For Wireless Applications by s.l.haridas and n.k.choudhari Minimum Transition Hybrid Register Exchange Method has require less power than the register exchange and Hybrid register exchange designs[13]. FPGA Implementation of Viterbi Decoder using Trace back Architecture by Swati Gupta,Rajesh Mehra RE becomes impractical due to its high power consumption and large routing overhead. Therefore it is not suited for low power applications such as wireless communications systems. One the other hand, Traceback (TB) traces back the maximum likelihood path starting from the best state. It traces back the survivor path after the entire code word has been received and generates the decoded output sequence [6].

V. DESIGN AND VLSI IMPLEMENTATION OF A LOW PROBABILITY OF ERROR VITERBI DECODER

C.Arun and V.Rajamani have presented paper on Design and VLSI implementation of a Low Probability of Error Viterbi decoder[12]. In which, for different constraint length they have measure probability of error. The authors obtained the results shown below:

Constraint length (K)	Conventional method (Polynomial)		Proposed method (Non polynomial)	
	d_{free}	Probability of error (P_e)	d_{free}	Probability of error (P_e)
3	10	$7.11*10^{(-9)}$	11	$8.12*10^{(-10)}$
4	13	$4.00*10^{(-11)}$	14	$5.13*10^{(-12)}$
5	16	$2.17*10^{(-13)}$	17	$3.19*10^{(-14)}$
6	18	$6.51*10^{(-15)}$	19	$1.23*10^{(-15)}$
7	20	$2.06*10^{(-16)}$	22	$7.1*10^{(-18)}$
8	22	$7.1*10^{(-18)}$	25	$4.11*10^{(-20)}$
9	24	$2.16*10^{(-19)}$	28	$2.32*10^{(-22)}$
10	27	$1.22*10^{(-21)}$	31	$1.34*10^{(-24)}$
11	29	$4.29*10^{(-23)}$	34	$7.4*10^{(-27)}$

Table 1: Author results

Finally, authors have concluded that the performance of Viterbi decoder when increases the constraint length the performance of Viterbi decoder is improves.

VI. CONCLUSION

In this paper, brief overview of implementation of Viterbi decoder using FPGA. The use of error-correction codes has proven to be an effective way to overcome data corruption in Digital communication channels. Register exchange method requires complex hardware compare to the Traceback method for larger constraint length though it will give faster speed. Thus, Traceback method is area efficient and better than RE method. The performance in AWGN channels improves as the constraint length increases.

REFERENCES

[1] S. K.Hasnain, Azam Beg and S. M. Ghazanfar Monir (2004).“Performance Analysis of Viterbi Decoder Using a DSP Technique”. IEEE. pp 201-206.
[2] Vasily P. Pribylov, Alexander I. Plyasunov (2005). “ A Convolutional Code Decoder Design Using Viterbi Algorithm with Register Exchange History Unit” .SIBCON. IEEE.
[3] Bernard Sklar, Digital Communications Fundamentals and Application, Published by Pearson education, Year 2003.
[4] B.P.Lathi, Modern Digital and Analog Communication Systems, Third Edition.
[5] A. J. Viterbi, “Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm,” IEEE Trans. Inform.Theory, vol. IT-13, pp. 260-269, Apr. 1967.

[6] Swati Gupta, Rajesh Mehra, "FPGA Implementation of Viterbi Decoder using Trace back Architecture" International Journal of Engineering Trends and Technology-May to June issue 2011.
[7] Behzad Mohmad Heravi and Bahram Honary, "Multi-rate Parameterized Viterbi Decoding for Partial Reconfiguration," PGNet 2006.
[8] Stefan Bitterlich and Heinrich Meyr (1993). "Efficient Scalable Architectures for Viterbi Decoders". Aachen University of Technology, Templergraben , Germany.pp 89-100.
[9] KIM MIM WOO (2005). "High-speed Viterbi Decoder Architecture".
[10] Yun-Nan Chang, Hiroshi Suzuki, and Keshab K. Parhi (2000). "A 2-Mb/s 256-State 10-mW Rate-1/3 Viterbi Decoder". IEEE Journal of Solid-State Circuits.Vol. 35 pp.826-833.
[11] Vasily P. Pribylov, Alexander I. Plyasunov (2005). "A Convolutional Code Decoder Design Using Viterbi Algorithm with Register Exchange History Unit".SIBCON. IEEE.
[12] C.Arun and V.Rajamani,"Design and VLSI implementation of a Low Probability of Error Viterbi decoder". Emerging Trends in Engineering and Technology, 2008. ICETET '08.
[13] S.L.Haridas and Dr.N.K.Choudhari,"A Low Power Viterbi Decoder Design With Minimum Transition Hybrid Register Exchange Processing For Wireless Applications", International journal of VLSI design & Communication Systems (VLSICS) Vol.1, No.4, December 2010.

