ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

VLSI Implementation of Trellis Encoding and Decoding Method for a Noise Robust Speech Recognition

Priya dharshini, G. and S Dr. Dhanalakshmi

Easwari Engineering College, Department of Electronics and Communication Engineering, Chennai – 89, India

ARTICLE INFO

Article history:
Received 10 March 2015
Received in revised from 20 March
Accepted 25 March 2015
Available online 10 April 2015

Keywords:
Noise robustness, Speech recognition, Spectral subtraction, Trellis encoding, Viterbi decoding.

ABSTRACT

Presence of noise in speech signals degrades the performance of automatic speech recognition systems. The spectral subtraction method is an effective method for the reduction of noise. In this method, an average signal spectrum and average noise spectrum are estimated, so that average signal-to-noise ratio (SNR) is improved. Spectral Subtraction is implemented by the estimation of the noise spectrum from regions that are estimated as "noise-only" and subtracting it from the rest of the noisy speech signal. It is assumed that the noise in the speech signal remains relatively constant. In this paper we have presented the hardware design implementation of low power Viterbi encoding and decoding algorithm for the cancellation of noise in speech signals. We have developed our own approach for the design of a 2/3 bit rate encoder and decoder. Convolution encoders are designed with state diagram methods instead of using shifters and adders. Viterbi decoder part includes Branch Metric Unit (BMU), Add Compare Select Unit (ACSU), Normalization Unit, Decision Unit and Output Unit have been implemented for achieving low power. The design has been implemented on Cyclone II FPGA using Quartus 9.0 software with supporting simulation tool Modelsim 6.4a. The results obtained from synthesis, simulation and hardware testing were precise and original information was recovered successfully.

To Cite This Article: Priya dharshini, G. and S Dr. Dhanalakshmi, VLSI Implementation of Trellis Encoding and Decoding Method for a Noise Robust Speech Recognition. ***Aust. J. Basic & Appl. Sci.,*** *9(15): 83-88, 2015*

INTRODUCTION

Many systems rely on Automatic Speech Recognition (ASR) to process the specific tasks. Using speech as the input it is important to ensure that the background noise will not degrade the systems performance. Spectral Subtraction (SS) is an algorithm which have the capability to reduce the degrading effects of noise acoustically added in speech signals. With the applications from speech and language development in young children to aiding individuals with hearing impairments, ASR is becoming more familiar and the demand for efficient systems is clearly seen. While humans are the best examples of ASR, the term we know it usually means the process in which a computer recognizes and identifies spoken words. There are so many tasks that involves interfacing with a computer can potentially use ASR, the following applications are as follows: Dictation, Command and Control, Telephony, and Medical/Disabilities.

This paper focuses on the removal of “white noise” in speech signals, which is used to enhance the speech and strengthens the performance of systems relying on ASR. White noise is a type of noise that is produced by combining sounds of all different frequencies together, because it contains all frequencies and white noise can drown or mask other sounds which may contain significant information needed for input into an ASR system. If white noise is present in a speech signal and when it is removed from that signal, then we can see an improvement in the quality of the speech and efficiency is improved for most of the ASR systems. Other methods used to reduce the amount of noise in speech signals include: Noise-cancelling microphones, although essential for extremely high noise environments such as the helicopter cockpit, offer little or no noise reduction. Another one of the most efficient techniques to improve robustness of speech recognition systems on additive noise consists in training the acoustic models with data corrupted by noise at different signal-to-noise ratios (SNR). However it is stated that this method requires training by individuals in different environments.

As our day to day life becomes more complicated, we find ourselves involved in many projects. Most of us require our hands or some other sort of physical interaction to communicate with others. ASR provides a “hands free” way to complete

Corresponding Author: Priya dharshini, G, Department of Electronics and Communication Engineering, Easwari Engineering College, Chennai - 89, India.
E-mail: dharshini3330@gmail.com

a variety of duties by simply speaking. ASR systems can optimize more tasks and complete them at a rate that is substantially faster. In addition to this, the systems can enhance the way the hearing impaired communicate, and it improves security and provides authentication for many applications. For these reasons, there is an obvious requirement for adequate ASR systems and their integration into our everyday life.

The main aim of this project is to introduce a forward error correction technique known as convolutional coding with Viterbi decoding. More particularly, we will focus primarily on the Viterbi decoding algorithm itself. This project will provide a detailed description of the algorithms for the generation of random binary data, convolutionally encoding the data and perform Viterbi decoding on the quantized channel symbols for the recovery of original binary data. The main aim of forward error correction (FEC) is to improve the capacity of a channel by adding some carefully designed redundant information to the data being transmitted through the channel. The process of adding the redundant information is known as channel coding. Convolutional coding and block coding are the two major forms of channel coding. Convolutional codes are used for serial data, one or a few bits at a time. Block codes are used on relatively large (typically, upto a couple of hundred bytes) message blocks. There are several useful convolutional and block codes, and a wide range of algorithms for decoding the received coded information sequences to recover the original data are present.

Overview of convolutional encoding and viterbi decoding:

Convolutional encoding with Viterbi decoding is a FEC technique that is particularly suited to a channel in which the transmitted signal is corrupted mainly by additive white gaussian noise (AWGN). We can assume AWGN as noise whose voltage distribution have characteristics that can be described using a Gaussian or statistical distribution. This voltage distribution which has zero mean and standard deviation is a function of the signal-to-noise ratio (SNR) of the received signal. Let's assume that the received signal level is fixed. Then if the Signal to noise ratio is high, the standard deviation of the noise is small. In digital communications, Signal to noise ratio is usually measured in terms of E_b/N_0, which stands for energy per bit divided by the one-sided noise density.

In this paper, a new approach to design Viterbi decoder using finite state machines instead of using shift registers and adders is proposed as FSM reduces both static and dynamic power. The Viterbi algorithm is the optimal solution for the convolutional codes. The nonlinear and recursive nature limits the maximum achievable throughput rate. The most common solution to develop a high throughput Viterbi decoder is fully parallel approach, where ACS units are assigned to each state with high radix structure. However, as the constraint length rises, the hardware complexity increases exponentially, also the power consumption. The first problem arised is the large number of ACS operations. The 2^{K-1} ACS operations are required for each iteration, where K is the constraint length. The second problem is implementing a high speed trace back unit is more difficult than that based on the register exchange method.

Encoding of convolutional codes can be accomplished using simple registers. In convolutional encoder, the message stream continuously runs through the encoder unlike in the block coding schemes where the message is first divided into long blocks and then encoded. Thus the convolutional encoder requires very little buffering and storage hardware. To generate the output, the encoder uses 7 values of the input signal (1 present input bit and 6 previous input bits). The set of values of input data in the shift register is called as state. The number of input data values which are used to generate the code is called the constraint length. Each set of outputs is generated by XORing a pattern of current and shifted values of input data.

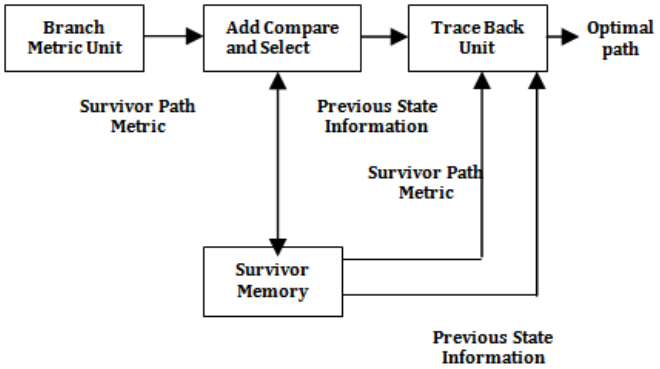

Fig. 1: Block diagram of Viterbi decoder.

The block diagram of Viterbi decoder is shown in Figure 1. When a sequence of data is received from the channel, it is necessary to estimate the original sequence that has been sent. The process of identifying the original message sequence from the received data can be done using the diagram called "trellis". A Viterbi decoder uses the Viterbi algorithm for decoding a bit stream that has been encoded using Forward error correction based on a convolutional code.

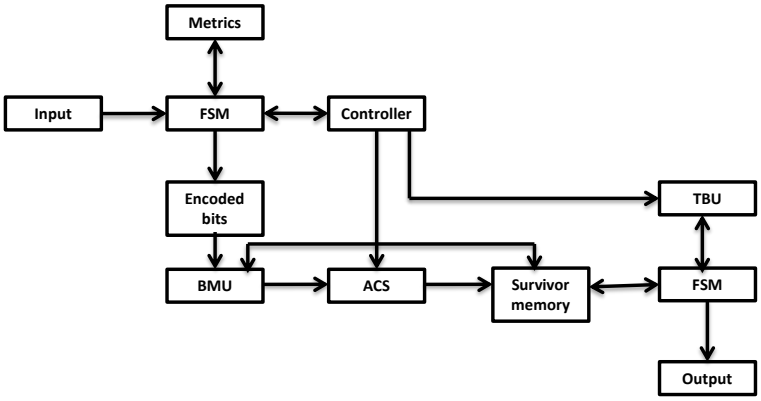

Fig. 2: Block diagram of the proposed algorithm**.**

Proposed algorithm:

In earlier designs, neural networks is used. In proposed design, Trellis Encoding and Decoding method is used.

The proposed algorithm is shown in Figure 2. It consists of input speech signal, metrics, finite state machine, Encoding the data, Branch metric unit, Add-compare-select unit, Survivor memory, Trace back unit, and output.

The function of branch metric unit is to calculate branch metrics, which are adjusted between every possible symbol in the alphabetic code, and the received symbol. There are hard decision and soft decision Viterbi decoders. A hard decision Viterbi decoder will receive a simple bit stream on its input, and Hamming distance is used as metrics. A soft decision Viterbi decoder will receive a bit stream which contains information about the reliability of each received symbol.

A path metric unit summarizes branch metrics to get metrics, where K is the constraint length of the code, which can eventually be chosen as optimal. Every clock it makes 2^{K-1} decision paths. The results are written to the memory of a trace back unit. The core elements of a Path Metric unit is Add-Compare-Select units. The way they are connected between themselves is defined by a specific code's trellis diagram. Since branch metrics are always ≥ 0, there must be an additional circuit preventing metric counters from overflow. An another 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 kept 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. The simplest 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. When it passes upward through each thresholds, a counter is incremented that reflects the "noise" present on the incoming signal.

Trace back unit restores maximum-likelihood path from the decisions made by PMU. Since it does in inverse direction, a viterbi decoder comprises a FILO (first-in-last-out) buffer to reconstruct it in correct order.

The steps for simulating a communication channel using convolutional encoding and Viterbi decoding are as follows :

1. Generate the data to be transmitted through the channel-result is binary data bits.
2. Convolutionally encode the data-result is channel symbols.
3. Map the one/zero channel symbols onto an antipodal baseband signal, which produces transmitted channel symbols.
4. Add noise to the transmitted channel symbols-result is received channel symbols.
5. Quantize the received channel levels-one bit quantization is called hard-decision, and 2 to n bit quantization is called soft-decision (n is usually three or four).
6. Perform Viterbi decoding on the quantized received channel symbols-result is again binary data bits.
7. Compare the decoded data bits to the transmitted data bits and count the number of errors.

Generating the data to be transmitted through the channel can be accomplished simply by using a random generator. The one that produces a uniform distribution of numbers on the interval 0 to a maximum value is provided in C: rand (). Using this

function, we can say that any value less than half of the maximum value is a zero; any value greater than or equal to half of the maximum value is a one.

Convolutionally encoding the data is accomplished using a shift register and associated combinatorial logic that performs modulo-two addition. A shift register is a chain of flip-flops wherein the output of the nth flip-flop is tied to the input of the (n+1)th flip-flop. Every time the active edge of the clock occurs, the input given to the flip-flop is clocked through the output, and thus the data is shifted over one stage. The combinatorial logic is obtained in the form of cascaded exclusive-or gates. This function uses exclusive-or gate with two-input, one-output gate often represented by the logic symbol and its truth table is given by Figure 4 and Table 1.

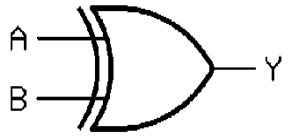

Fig. 3: Symbol of a NAND gate.

Table 1: Truth table of a NAND gate.

Input A	Input B	Output Y
0	0	0
0	1	1
1	0	1
1	1	0

The exclusive-or gate performs modulo-2 addition of its inputs. When we cascade two-input exclusive-or gates, with the output of the first one fed to the second, the output of the second is fed to one of the inputs to the third one, etc., the output of the last one is the modulo-2 sum of the inputs. Assume that we have the two basic components of the convolutional encoder (flip-flops comprising the shift register and exclusive-or gates comprising the associated modulo-two adders) defined, let's look at figure 3- convolutional encoder for a rate 1/2, K = 3, m = 2 code:

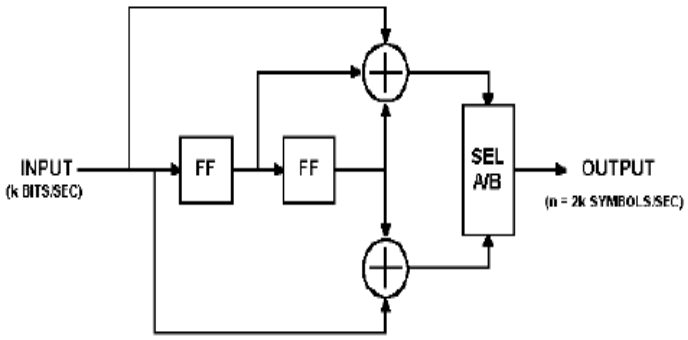

Fig. 4: Convolutional Encoder.

In this encoder, data bits are given at a rate of k bits per second. Channel symbols are the output at a rate of n = 2k symbols per second. The input bit is stable during the encoding cycle. The encoding cycle starts when an input clock edge starts. When the input clock edge initiates, the output of the left-side flip-flop is clocked into the right- side flip-flop, the previous input bit is clocked into the left- side flip-flop, and a new input bit will be available. Then the outputs of the upper and lower modulo-2 adders become stable. The output selector (SEL A/B block) cycles through two states in the first state, it selects the output of the upper modulo-2 adder; in the second state, it selects the output of the lower modulo-2 adder.

The convolutional encoder shown above encodes the K = 3, (7, 5) convolutional code. The octal numbers 7 and 5 represent the code generator polynomials, which when read in binary (1112 and 1012) correspond to the shift register connections to the upper and lower modulo-two adders, respectively. This code has been determined to be the "best" code for rate 1/2, K = 3.

Adding noise to the transmitted channel symbols produced by the convolutional encoder involves generating Gaussian random numbers, scaling the numbers according to the desired energy per symbol to noise density ratio, Es/N 0, and adding the scaled Gaussian random numbers to the channel symbol values.

Implementation:

Implementation of Convolution Encoder and Viterbi Decoder for constraint length 7 and bit rate 1/2 is done using Verilog HDL. The design is simulated and synthesized using ModelSim Altera web edition 6.4a. The design is implemented and the

power analysis report is obtained using Altera Quartus II.

Experimental results:

The Convolution Encoder and Viterbi Decoder for constraint length 7 and bit rate 1/2 has been developed and synthesis is done. Figure 5 shows the output of an Encoder. Figure 6 shows the simulation output of a Viterbi Decoder. Figure 7 shows the RTL Schematic View of the Design.

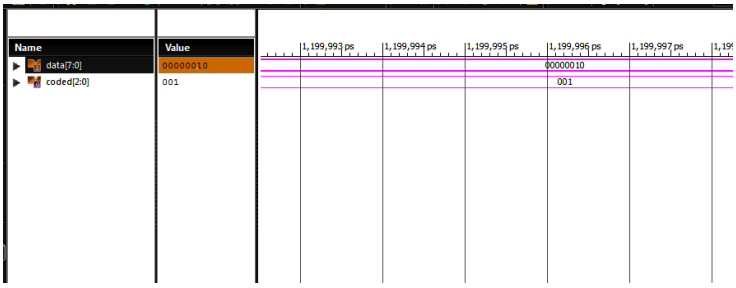

Fig. 5: Output of an Encoder.

The Simulation output of an encoder shows the time consumption.

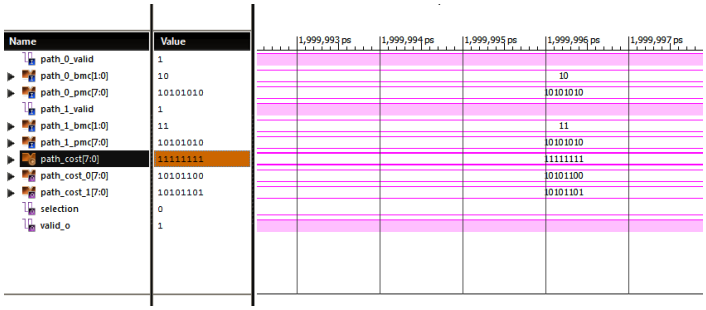

Fig. 6: Output of Add Compare and Select unit.

Table 2: Comparison Table for Existing and Proposed System.

Parameters	Area	Power
Deep Neural Networks (Existing system)	1443 (Number of logic elements)	310.27mW
Trellis Encoding and Decoding method (Proposed System)	1065 (Number of logic elements)	197.91mW

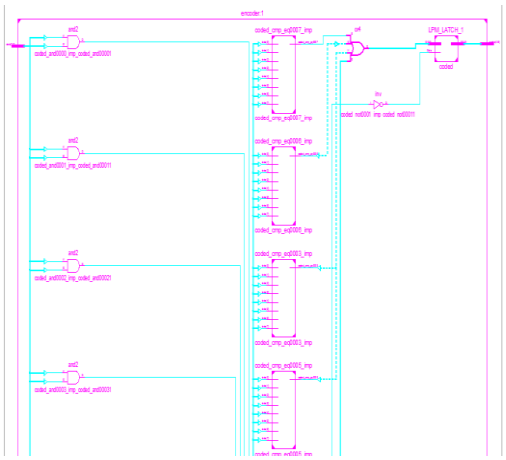

Fig. 7: RTL Schematic View.

The Simulation of the Add Compare and Select unit shows the time consumption.

The Comparison between the amount of power and area used for the existing and proposed system is given by Table 2.

The Architecture is shown using the RTL schematic View.

Conclusion:

Trellis Algorithm allows safe data transmission via error correction and original message can be recovered accurately without any noise. It was concluded from research that if trace back is started after going deeper into trellis diagram then more accurate data and low power can be achieved. Trellis algorithm of any rate can be designed using same basic principles and techniques.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Dr. S. Dhanalakshmi for her guidance in the course of my research.

REFERENCES

Bo, Li., Student Member and Khe Chai Sim, 2014. A Spectral Masking Approach to Noise-Robust Speech Recognition Using Deep Neural Networks. IEEE/ACM Transactions On Audio, Speech, And Language Processing, Vol. 22, No. 8.

Boldt, J., 2011. Binary masking & speech intelligibility. Ph.D. dissertation, Aalborg Univ., Aalborg, Denmark.

Brown, G.J., 1992. Computational auditory scene analysis: A representational approach. Ph.D. dissertation, University of Sheffield, Sheffield, U.K.

Hinton, G.E., L. Deng, D. Yu, G.E. Dahl, A.R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen and T. N. Sainath, 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Process. Mag., 29(6): 82–97.

Hinton, G.E., S. Osindero and Y.W. The, 2006. A fast learning algorithm for deep belief nets. Neural Comput, 18(7): 1527–1554.

Li, B. and K.C. Sim, 2013. Noise adaptive front-end normalization based on vector taylor series for deep neural networks in robust speech recognition. in Proc. ICASSP, pp: 7408–7412.

Li, B., Y. Tsao and K.C. Sim, 2013. An investigation of spectral restoration algorithms for deep neural networks based noise robust speech recognition. in Proc. Interspeech, pp: 3002–3006, ISCA.

Lyon, R., 1983. A computational model of binaural localization and separation. in Proc. ICASSP, pp: 1148–1151.

Maas, A.L. and Q.V. Le, 2012. Recurrent neural networks for noise reduction in robust ASR. in Proc. Interspeech, pp: 22–25, ISCA.

Rennie, S.J., P. Fousek and P.L. Dognin, 2012. Factorial hidden restricted Boltzmann machines for noise robust speech recognition. in Proc. ICASSP, pp: 4297–4300.

Seltzer, M.L., D. Yu and Y.Q. Wang, 2013. An investigation of deep neural networks for noise robust speech recognition, in Proc. ICASSP, pp: 7398–7402.

Vinyals, O., S.V. Ravuri and D. Povey, 2012. Revisiting recurrent neural networks for robust ASR. in Proc. ICASSP, pp: 4085–4088.

Wang, D.L. and G.J. Brown, 1999. Separation of speech from interfering sounds based on oscillatory correlation. IEEE Trans. Neural Netw., 10(3): 684–697.

Wang, D.L. and G.J. Brown, 2006. Computational auditory scene analysis: Principles, algorithms and applications. NewYork, NY, USA: Wiley/IEEE.