IMPLEMENTATION OF HSLP VITERBI DECODER DESIGN FOR TCM DECODERS

Prabhu Kiran¹, David Solomon Raju Y²
¹ PG Student, M. Tech, Dept of ECE Holy Mary Institute of Technology & Science, A.P, India
²Assoc. Prof. ECE, Holy Mary Institute of Technology & Science, A.P, India

Abstract: - The aim of the project is execution of Viterbi algorithm applying HDL coding. The overall objective is to clearly understand the Hidden Markov model and Viterbi Decoder, to evaluate the basic functionalities and steps involved in Viterbi algorithm, to project on the implementation of Viterbi algorithm through HDL code and finally to critically analyze the results obtained through HDL code. The main purpose of this study is to yield the gains obtained by the developers with the usage of Viterbi algorithm. This project not only helps the students related to the communications but it also helps the people who are in the field of decoders as it is one of the efficient method for reducing the errors while communication procedure is in advance. Here, HDL code is used in order to implement the Viterbi algorithm in a proper way. Apart from students and the business people, one can easily understand and analyze the Viterbi algorithm concepts and can gain more knowledge on the tools that are used in this project.

Keywords: - Viterbi algorithm, HMM, Branch metric generation, State metric generation, Chain back, decoder, VD.

I. 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 necessary for the telecommunication to reduce the data corruption by providing a suitable solution to the errors occurred in the communication process. One such method that decodes the process by simultaneously correcting the process effectively is Viterbi algorithm. For decoding the convolution codes Viterbi algorithm is the highest recognizable algorithm. This algorithm may be described with software as well as hardware implementations. To engage well organized communications an efficient data is presented by the digital systems. Data corruptions are the important issue confronted by the digital communication systems. To decrease data corruptions error correcting codes is a best technique. Almost all communication systems followed it because it’s power to decode efficiently, even Viterbi algorithm needs very typical hardware. While the decoding operation is in advance, the functioning obstructions can be eliminated, So that an improved method, Adaptive Viterbi Algorithm is used. The decoding of codes can be done very fast, as this algorithm is very effective in high speed functions. Convolution codes are used to gain a possible code sequences AVA uses maximum-likelihood decoding process.

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. A third generation wireless cellular standard, under preparation, plans to adopt turbo coding, which stems from convolutional coding. The Viterbi decoding algorithm, proposed in 1967 by Viterbi, is a decoding process for convolutional codes in memory-less noise. The algorithm can be applied to a host of problems encountered in the design of communication systems. The Viterbi decoding algorithm provides both a maximum-likelihood and a maximum a posteriori algorithm. A maximum a posteriori algorithm identifies a code word that maximizes the conditional probability of the decoded code word against the received code word, in contrast a maximum likelihood algorithm identifies a code word that maximizes the conditional probability of the received code word against the decoded code word. The two algorithms give the same results when the source information has a uniform distribution.

The above mentioned works show that their design substantially reduces power dissipation of Viterbi decoders. Two major techniques, clock gating and toggle filtering, were investigated in this thesis. In our experiments, estimated power dissipation was estimated on the basis of the switching activity measured through behavioral simulation. Experimental results indicate that our methods effectively reduce the power dissipation of Viterbi decoders.

Hardware description language called Verilog HDL is used to validate this project, where it is one of the hardware descriptive languages that stand for Verilog Hardware Description Language. This project makes use of two main tools namely Modlesim-simulation and Xilinx-ISE-synthesis for successfully reaching its objectives. Further of this project provides a clear description on Adaptive Viterbi Algorithm, its execution process and various kinds of languages and tools for evaluating the Viterbi Algorithm.
The project mainly carries with the decoder problem while generating the coding. The Viterbi algorithm is used for finding the Viterbi path in a sequence which is a VLSI implementation of the Viterbi decoder. This algorithm is employed in high speed and high accuracy functions such as, WIFI, mobile technology like GPRS and satellite communications, broadband, wireless technology like WIMAX.

The project identifies the routes in the trellis diagram that are named in the Viterbi decoding. The Viterbi algorithms are mainly employed in the digital communications, decoding the Convolutional codes and the image and speech recognition. With this project, one can easily estimate the importance of Viterbi algorithm in the reducing the decode errors in the communication and can even make use if this project to create the new algorithm which provides better features than the Viterbi algorithm. Hence, from this context it can be expressed that this project is mostly useful for the various students related to the communication filed and the organization in raising their levels in the global market by reducing their errors in the communication process.

This section describes the introduction of the project concepts. The Literature review i.e. background on the operation of convolutional encoders and Viterbi decoders is provided in section II. A brief description of low-power design techniques investigated in this thesis is also covered in this chapter. Section III proposes implementation of Viterbi decoders. Section IV describes the analysis environment and Section V lists experiment results, finally Section VI summarizes with the overall conclusion and future scope followed with references.

II. LITERATURE REVIEW

Viterbi algorithm is an approach towards finding the most common sequence of hidden states in all listed states. It is dynamic programming algorithms that find the probability of all observed sequence for each combination. Pr (observed sequence hidden state combination) It is a feasible procedure to find the common sequence .The complete calculation in each combination is much costly .It is evaluated for the error correction for noise in the digital communications. Viterbi algorithm is familiar algorithm works on the state machine assumption for the conventional codes. By using the system can be modeled at certain state. There are finite numbers of states. There will be a survivor path mostly a common path in a multiple sequence path that can lead to a given state.

It can describe the hardware and the soft ware implementations. The noisy channels are usually corrected by the conventional codes as they are efficient for correcting the corrupted channels. Satellite communications, CDMA and GSM cellular, dial modem, deep-space communications and 802.11 wireless LANs. Mostly use the conventional codes. Information theory, speech theory, keyword spotting, computational linguistics and bioinformatics use this algorithm usually. The algorithm is not more likely i.e., it may create numerable statements. In the first step both the observed events and the hidden events must be within the same sequence and that sequence must resemble the time. While comes to the next step the two sequences must be put together and the known or the observed events must resemble the accurate one hidden event.

The next coming third step computing the most probable hidden sequence up to certain point “t” depends on the absorbed point within the sequence at point “t-1”. The algorithm examines the forward by moving to new set of states by combining the metric of possible previous states with the incremental metric of transition due to the event and select the best for a event occurred. In many cases the state transition graph is not connected fully. This algorithm can relate the active programming that discovers the single most probable observed sequence. Sometimes the statically parsing active programming can be used to detect the single most common context-free derivation of a string. After all compounding of the incremental metric and the state metric computing only the best lasts and all other paths are disposed. In iterative Viterbi decoding one may find the sequence of engaged that corresponds the right test for a given HMM.

The working of the iterative Viterbi algorithm is by iteratively raising an altered Viterbi algorithm, estimating the score for filler till intersection An flip-flop algorithm, called Lazy Viterbi algorithm, is recently projected that works not done by expanding the nodes until it really needs to, usually manages to get away than the normal Viterbi algorithm. It has been continued to control in a settled finite automation in gild to have speed operational for the generate trellis with the state transition pointing the variable amount of history. In many situations the history completes as the state machine starts at a well-known state as paths are kept at the sufficient memory. When there is a case of bounded resources as one exercise in convolution encoding the decoder need to leave the history for the sake of acceptable performance level. There are alterations that made easy for the computational load and the memory elements that lean to maintain constant.

A. Errors occurred during the coding in communication process

Communication is a process of transferring data from one person to other person involves a lot of coding during the programming of its mechanism and the probability of getting errors is ample. Some of the simple bit errors can be adjusted by interpreting the real bit sequence during communicative path is down line. Most of the arbitrarily problems can be solved randomly utilizing few of the
important features of Viterbi algorithm which yields to the original sequence. The most important feature for communication is to facilitate error free data transmission among digital or analog functioning signals along with amplification. Coding in communication system is basically categorized into four sections as

- Encryption: mainly used for security of data,
- data compression: used for data streaming and to reduce the space,
- Data translation: used to demonstrate the data for transmission of communication channels
- Error Control: identifies the errors and correct them as soon as possible.

For digital signals data represented as 0 and 1's, so as to detect errors and analyzing the noise occurred while transmission as well as to correct those errors. In normal cables the error is due to the random motion and some deviation occurs when conduction is through various components like resistors, capacitors, inductors etc, which is known well known as thermal noise. This is one of the major sources of noise for cable communication system. If there are various sources of noise in wireless communication systems such as in mobile phones, disturbances are of other user signal noise interruption. The original signal is normally added with the noise signal at receiver input. The forward error correction (FEC), auto repeat request (ARQ), hybrid ARQ and error code correction (channel coding) are the general methods used for error correction. As the error in coding generally occur due to storage or transmission for protection of digital data is with the modern internet communication and telecommunications. The rich mathematical theory is also employed number of error protection techniques.

B. Proposed Method

Proposed is a probe of new facts that are exercised in this project. General method is an organized engineered which will determine the problems, suggest solutions and finally prepares the gathered data. For, project the data need to be gathered from many resources where the concept will identifies proves to be collected and the techniques that need to applied in the project. Generally there exist two methods for gathering the accurate data to the project. They are primary type and secondary type. In the primary type we need to gather the data manually without referring or taking the ideas from other papers where as in the secondary type we gather the data from numerous resources by referring the journals, magazines, books, etc. For the present project, it’s better to prefer the secondary resources as this project deals with the implementation of Viterbi algorithm. Here, the project cannot depend only on primary data as we will not find data by interviewing of surveying the people as all the people cannot know about this algorithm.

C. Proposed solution for the problem: Viterbi Algorithm

Wide range applications of the Viterbi algorithm are towards the DNA analysis, speech appreciation for cell phones communication and facilitates. The outcome of backtracks from all the branches may obtain the algorithm task. The Viterbi algorithm can perform step-by-step function as illustrated:

1) Initialization: Arrange all metric in the perfect format.
2) Computation step j+1: Suppose the previous step and use to identify the basic survivor paths for storage in all the states.
3) Final step Continue to compute the entire pending algorithm reaches with all-zero state like hood paths. Viterbi algorithm is most likelihood detected sequence with the MLSD with in all the inter-symbol interference (ISI) as well as memory less noise considering all the input state channel as well as observable sequence.

Let the Hidden Markov Model(HMM) with the states may be Y, at initial stage probabilities p I of being in state i and transition probabilities a of transitioning from state i to state j. Say we observe outputs. The state sequence most likely to have produced the observations is given by the recurrence relations.

\[ V_{sk} = P \left( \frac{X_k}{k} \right). \pi_k \]  
\[ V_{t,k} = \left( P(x_{t,j}) \frac{p_{j,k}}{k} \right). \max_y \]

Here \( V_{sk} \) is the probability of the most probable state sequence responsible for the first \( t+1 \) observations (we add one because indexing started at 0) that has \( k \) as its final state. The Viterbi path can be retrieved by saving back pointers which remember which state \( y \) was used in the second equation. Let \( P_t(k,t) \) be the function that returns the value of \( y \) used to compute \( V_t, k \) if \( t > 0 \), or \( k \) if \( t = 0 \).

D. Hidden Markov model and Viterbi decoder

Hidden Markova model: The chain of Markov is generally absorbed in noise processing signals. Markov chain is symbolized as \( \{X_k\}_{k \geq 0} \), hear \( k \) is basically an integer index. So as to quit the finite set that is for making secreted, Markov chain is hidden and can’t be observed in arbitrary state, thus it is experimentional known to be as stochastic process \( \{Y_k\}_{k \geq 0} \) this is an another linked process, as \( Y_k \) is governed with the Markov chain in the distribution links [14]. This hidden Markova model is known to be a bivariate discrete time process \( \{X_k, Y_k\}_{k \geq 0} \), where \( \{X_k\}, \{Y_k\} \) are the sequence of random independent variables as \( \{X_k\} \) is the Markov chain and conditional distribution of \( Y_k \). The hidden Markova model (HMM) is a signal facilitates to communicate with speech signals which achieved acceptance from almost all the communication systems.
The hidden Markov models consist of two classic layers sub cellular location known as upper layer and the functional class, which is lower layer. If any process is undertaken in the hidden Markov model the doubly stochastic process can’t be observed directly since, it is hidden and may be observed only with another stochastic process which will facilitates in sequential observation.

The two layers upper layer and lower layer are joined for analyzing multiple paths for the flow from begin to end. Nodes present at the ends of two layers encode the standards which are randomly hidden from the upper layer namely location class variables with the lower that is functional class variables. The direction of arrows present in between two layers is the transition flow indication where colors and shades are indicated as per the estimated probability counts based on training sequence.

Viterbi decoder: In general Viterbi decoder apparatus Viterbi algorithm mainly for decoding as well as encode fragment flow by using he forward error correction (FEC) intricacy encoding system. Viterbi decoder is mainly employed for encoding the convolutional data as it is able to overcome number of errors received at the input data due to channel noise. The Viterbi decoding algorithm is a state of the art algorithm used to decode convolutional binary codes (viewed as a trellis tree) used in communication standards (like Qualcomm’s CDMA standard). In the implementation of input code symbol stream this Viterbi decoder is used to operate in decoding with some likely sequence. Viterbi algorithm follows the most likely path for maximum encoders and decoders with three main processing steps which are listed below:

- Branch metric generation
- State metric generation
- Chain back

Before implementing the Viterbi algorithm it is essential to collect and relate all the noise with the Markov process in definite order. Viterbi detector includes the ISI channels having the predetermined memory noise driven with the MLSD and MAP sequence detector is utilized. Some of the important features of Viterbi decoder as listed below:

E. Block Diagram of Viterbi algorithm

The Viterbi algorithm is one of the standard sections in number of high-speed modems of the process for information infrastructure applicable in modern world. The dynamic algorithm includes some path metrics so as to compute the path sequence transmitted earlier the name Viterbi algorithm arrived after Andrew Viterbi and is represented as VA for reorganization, record of huge possibility decodes as well as least reserved decoding are generally similar in a defined binary symmetric channel. The basic performance of the Viterbi decoder is analyzed with the block diagram shown below. It consists of three main blocks branch metric unit, add compare select and trace back unit. The unit of branch metric will calculate all the branch metrics and then processed to add compare for selecting the surviving branches as per the branch metrics finally the decoded data bits are generated by the trace back unit.

The overall performance of the Viterbi algorithm is analyzed with the help of conventional codes. The simulated block diagram explains the operation of detecting and correcting the coding errors in normal communication system. The transmitted bits of data are encoded in the first block with conventional code that is (CC encoder) which are modulated by means of binary pulse-amplitude modulation (PAM) so as to tune those bits into antipodal bits and process to the additive white Gaussian noise (AWGN) channel thus obtained data combined with noise is supplied to soft decision Viterbi algorithm (SDVA) which only accepts the antipodal data at the input for decoding and produces the output decoded bits.

Implementation of the Viterbi algorithm is supported with two main steps the initial step is to select the trellis from the bits that are achieved at the input at the receiver. A simple trellis figure shows with four stage points for transmission, each state is represented with a dot and the state transition is shown as edge of branch. Each and
every branch is known as the branch metric as it is associated at Euclidean distance with the symbol towards final transition.

\[
BM (rr, bb) = (r0-b0)^2 + (r1+b1)^2
\]

\[
= r0^2 - 2r0b0 + b0^2 + r1^2 - 2r1b1 + b1^2 = r0b0 + r1b1
\]

Where, \( rr = \) symbol received at the input, \( bb = \) branch symbol. Both \( rr \) and \( bb \) are dependent on the used for conventional encoder. Under the basic assumption that there is no noise in the data and the value of \( r \) and \( b \) will vary between -1 to +1, the range of branch metric will range within -2 to +2. In case \( rr = bb \) branch metric would be 2, Similarly \( r0 = -b0 \) as well as \( r1 = -b1 \) and \( BM = -2 \). The path metric (\( \lambda \)) in the minimum Euclidean distance in the trellis does not required the actual value the original order of the floating point pair numbers is

\[
\lambda_{\text{new}} = \lambda_{\text{prev}} + r0b0 + r1b1
\]

The path metric \( \lambda \) is the shortest distance among cumulative state, thus distance of the path (Euclidean distance) is inversely proportional to the branch metric. After complication of generating a trellis it is necessary to find survivor path with maximum path metric. In the above the solid black line is the survivor path.

**F. Description of Viterbi Algorithm**

Viterbi algorithm is basically implemented to decode the errors found in convolution encoded sequence. As discussed the Viterbi algorithm will make use of trellis structure in finding the coded sequence based on the transmission signals. Since each and every code sequence will follow based on the trellis process of encoding data. Considering an example of trellis diagram of half rate, three convolution encoder \( K=3 \) and 15 bit messages with four possible states shown in 4 horizontal rows with dotes.

![Figure 3: Viterbi algorithm trellis](image)

For calculating the branch metric can be obtained with the trellis using the Euclidean analysis as follows:

\[
BM (rr, bb) = (r0-b0)^2 + (r1+b1)^2
\]

\[
= r0^2 - 2r0b0 + b0^2 + r1^2 - 2r1b1 + b1^2 = r0b0 + r1b1
\]

**Figure 4: Trellis diagram of Viterbi algorithm.**

18 columns shows the time instants from \( t=0 \) to \( t=17 \) both \( t=0 \) and \( t=17 \) has the four dot column which is initial and final state situations while encoding messages. The state transition is shown with a dotted line at zero input. The figure shows the state of trellis, which reach the encoding of messages.

![Figure 4: Trellis diagram of Viterbi algorithm.](image)

After complication of generating a trellis it is necessary to find survivor path with maximum path metric. In the above the solid black line is the survivor path.

Maximum likelihood (ML) sequence will be obtained from the track of paths that occur for Viterbi algorithm is essential for processing information coding. The common aid to analyze the Viterbi algorithm is the trellis diagram. It is the decoding algorithm used with convolution code. The input, output, receiver and error details are shown at bottom to the figure. The receiver pair of the channel after collecting the complete information regarding the process the Viterbi decoder is ready to function with the bit that are to be transmission following some steps. Initially number of state is to be selected at a negligible collected error metric then save the number state arrived. Accurate performs of step from the initial trellis is to be achieved. All the state sequence state numbers are to be saved. Work forward with record of all the selected states which
are saved in the previous steps which are to be built in by all the encoded convolution encoder.

G. Advantages of Viterbi algorithm

All the trellis occurred are arbitrarily solved even in presence of two or more simple errors in input string using Viterbi algorithm. At the same time the in presence of more errors the decryption will be low even then the algorithm works effectively this is the main advantage in implementation of Viterbi algorithm. The usage of this Viterbi algorithm is found to be advantageous due to its cost effectiveness in modulated minimize at the same time the functional performance in some situation would modulate in maintaining the original cost. Emerging linear functioning of linear pulse distance is due to convenient source sequence. In the analysis of Viterbi algorithm basic concept is to know the maximum-likelihood sequence detector (MLSD), MAP sequence required for the inter-symbol interference (ISI) channels which has some noise in the statistics of signal. Importance and working of Viterbi algorithm can be understood from the description and block diagram explanation of Viterbi algorithm.

III. IMPLEMENTATION

There are basically two Viterbi algorithms namely isolated sign language Viterbi algorithm and continuous sign language Viterbi algorithm both are standard used to search the frame simultaneously. To find the sequence of hidden states which is called as Viterbi path the Viterbi algorithm is used which is a dynamic programming algorithm. A state machine assumption is used for the functioning of Viterbi algorithm. There is finite number of states, at any time system being modeled in some state. The survivor path which is at least one of the most likely paths to the state when number of sequences of paths can be directed to given state. The most likely state is kept by examining all the possible states which are the fundamental assumption of the algorithm. Thus by keeping only one path is necessary and do not need to keep all the track of all states. This is the first assumption. A new path from the previous state is marked by additive metric which is the second assumption. And the third assumption is that in some sense events are accumulative over a state. By moving advance in a new state it chooses the best by combining the additive metric with the previous path an new set of stated can be examined by the algorithm whenever an event occurs. The transition property from old path to new path is linked with the additive metric. Let us consider an example for this. It is only possible to beam half of the symbols from even numbered path and the other half of the states from odd numbered path in data communications.

The state transition graph is not fully connected in almost all cases. To find the sequence of hidden states which is called as Viterbi path the Viterbi algorithm is used which is a dynamic programming algorithm. The Viterbi coder implemented is based on a 16-state rate 1/2 convolution coder with the following system equations:

\[ G_0(n) = x(n) + x(n-1) + x(n-3) + x(n-4) \]
\[ G_1(n) = x(n) + x(n-2) + x(n-3) + x(n-4) \]

Where \( x(n) \) is the un-coded input and \( G_0(n), G_1(n) \) are the encoded outputs. To implement the Viterbi decoder we will use a 16-state trellis diagram.

In digital signal processing, the decoder circuit use Viterbi algorithm as a basic conventional coding. The Viterbi decoder operates at very high speed decode processing to get desired result. The designing of Viterbi decoder needs large circuitry for calculations and memory allocation for results. The working of Viterbi algorithm is, it takes a tray of bits and try to get a direction in the trellis diagram with an output digit sequence which is similarly as the received sequence input bits. In trellis diagram each branch maintains a particular value. This value is known as the branch metric. If the weights are considered as branch metric values, the aim of the Viterbi algorithm is to calculate the least weight path from the left final column to the right most column of the trellis state diagram. At each branch node only few small path metrics can be omitted. At each node Viterbi algorithm computes a desired decision path value and an input label for each node. The decision value is taken from the last stage of computation and it is the smallest path value in that stage. From this observation we can conclude that the result of Viterbi algorithm is the smallest path metric previous computation stage.

Viterbi algorithm uses the software of Viterbi decoder for implementation. Initially it is essential to develop some data structure around the algorithm since there are implemented as arrays, thus the structure of most important six array is explained in following steps. Prepare the convolution encoder copy it to the next table for state transition encoder table. Before the process of decoding is stated the array is to be initialized in every step. Prepare the output table for convolution encoder. The array or table is represented for the present stage as well as next stage of convolution encoder, based on the high or low input values the next state is processed to show the present states. This array/table is called as the input table. Store each state history of all encoder for K at received pairs to symbols. The accumulated error array is to be stored in the form of metrics with add compare-select operation. This is known to be as accumulated error metric array.
Viterbi Decoder Architecture

The Viterbi decoder is separated as three functional units. The first part is an add-compare-select (ACS) unit designed for calculating the path metrics. The second unit is used for survivor memory management. It is the last sequence as known as survivor sequences and the survivor memory is used for storing survivor sequences. Register. The register exchange technique or the trace-back techniques are used for survivor memory implementation. The trace-back technique is also used in Technical Data Freeway’s (TDF) VHDL model of the Viterbi decoder. The third part of the Viterbi decoder is ACS (Add-Compare-Select Cell) unit. The ACS cells work is to lead together the branch metrics to the next attempting path metrics to choose the path that is less. The ACS cells also produce the pointer to the decision value which is the smallest path metric in the previous state and stored in the survivor memory.

The ACS functional unit has more number of ACS cells, and the number of cells is calculated by the parameters of ACS_UNITs and path metrics. While in decoding process registers and temporary registers are used for storing the parameters and path metrics. The ACS unit also computes the number of the input branches. The signal clock counter in the Viterbi architecture is used to control the computations of the decision values and path metrics and also drive the SMU control unit. The control unit carries two computations. In trellis diagram it calculates the states of upper part, in be twin lines with the path metrics in the lower part are parallel. The incoming clock counter counts from 0 to ACS_period-1. Temporary registers store the computed path metrics when the clock counter is greater than zero and when it is equal to zero path metrics registers are used for storing the finishing computed path metrics and the elements of the temporary registers. A small value should be chosen as the path metric register word length in order to avoid overflow. If generally used path metric is considered then the smallest path value in the previous stage also is selected. The ACS unit will stop processing when freeze signal will rise because of computing the whole burst in the ACS unit. This type of operation leads Viterbi decoder in the trace-back mode.

SMU control unit

To store the final sequences in the Viterbi decoder the survivor memory is used and controlled by the SMU unit. It is also based on trace back method. In this unit the alteration of studying the new branch metrics and trace back is scheduled by the state machine. The SMU control unit read the smallest path values or previous stage smallest values from the registers and outputs the decoded bits. The decision values are written in bursts at survivor memory when the branch metrics are inserted in bursts. It contains four counters. The first one is the ACS counter, used to count from 0 to ACS period-1 during the ACS operation. It generates the write-enable and load-enable signals. For counting from 0 to burst_length load counter is used in ACS operation. It generates the internal control signals and counts the input data. In trace back mode TB counter is used for counting from 0 to mem_size-1. Viterbi algorithm is the one of the simple and unique method and it has a wide spread in the complexity at the transmission side [28].

Viterbi algorithm is based on the strongly connected trills decoding of the binary convolution codes. The use of the error correcting codes has solved to be the effectively to overcome the data corruption in digital signal communications. With the utmost likelihood Viterbi algorithm for the better decoding of the convolution codes. It also the easy way for short constraints to decode. Viterbi algorithm implements the ML coding by reducing the complexity. It eliminates the east trellis paths at each level of transaction. It gets efficient via focus on the survival of the trellis. The Viterbi decoder makes use of Viterbi algorithm for decoding of the bit stream that is encoded with the forward error correction based on the code. While coming to the implementation parts there are some key points that are to be remembered that are:

- To generate the required data that is to be transmitted through channel can be achieved with a random number generator.
- Using the shift registers and the convolution logics that perform the modulo-two addition the convolution encoding of data is possible.
- Mapping the one / zero output of the convolution encoder onto the antipodal baseband and signaling scheme that produce the transmitted channel symbols.
- Addition of noise channels to the transmitted channels symbols by the encoder that implies in the generation of the Gaussian random numbers to produce the received channel symbols.
- A Viterbi algorithm decoder should precisely work on the infinite or at least the floating point numbers.
- Result again the binary data bits on the quantized received channel to perform the Viterbi decoding.
- Comparison of the decoded data to the transmitted data bits and the count number of errors.

The Simple and the unique as described Viterbi algorithm operates on the block of received data and the block length decides the decoding speed. The implementation of this design is done on Xilinx Spartan 2e xc5s100-5eq208. Using Xilinx tool Viterbi decoder is first
synthesized and simulated for the proposed pipeline structure. Trace back method (TB) and register exchange method are the two techniques used for decoding the data.

IV. ANALYSIS

Viterbi algorithm is extensively used decoder technique in digital communication systems because it secures the information message from the affects like noise, fading when transmitted from sender to receiver. Viterbi algorithm considers the regularly used symbols if the receiver gets the damaged sequence of symbols from the transmitter.

A. Analyzing the results obtained through VHDL coding for the implementation of Viterbi Algorithm

The implementation of Viterbi algorithm is somewhat difficult even though the process of algorithm is simple. Conventional encoding can be easily implemented on Viterbi algorithm even though there exist a large gap in complexity with the transmission side. State trellis uses conventional encoding, the decoder explores rotates between states because it is a finite state machine. It requires large memory registers for storing results. There is some delay in final decision on a sequence of transitions because of the size of the input code is very high. By observing the transition metrics between states the decision can be done and the results are updated in the form of Hamming distance or Euclidean with the error-corrupted received sequence.

The computation process of conventional codes depends on the minimum distance and then on the constraint size and coding speed. There are continuous changes in implementation process in order to increase the parameters like Gain. Complexity raise up to maximum limit in order change the old techniques. Now a day’s some algorithmic part and Systolic architectures in implementing on various devices Adaptive Viterbi decoding (AVD). Only subsets of the states are used in AVD algorithm for storing and implementation process. Due to decrease in size of states, the performance also reduces. In order to increase the performance of decoder, it is implemented at shorter distances. When channel status information is available, Replace stronger codes with simpler codes for high speed switching between different coding rates.

B. Architecture of the programmable VD

The computation process of Viterbi decoder rises exponentially with the constraint size K. For raising size of k more hardware circuits are required for both treating power and memory .the reuse of resources for reducing the area occupancy and the operating frequency required by the UMTS standard is of primary concern. Three-bit soft decision has been adopted as quantized input: this alternative option represents a good trade-off compare complexity and accuracy. The Viterbi decoder architecture mainly consists of three units.

1. Branch Metric Unit (BMU): Calculation of the minimal length between the input pairs of bits and all the possible “ideal” pairs
2. Path Metrics Calculation: For each decoder state, calculating for survivor ending in a metrics needed. Hence to obtain the survivor with the minimal metrics are to be noted.
3. Trace back: To obtain the desired results this is responsible step is to be simulated at the hardware implementations that don’t store the actual information regarding the survivor paths. But are stores one bit of information every time

![Figure 5: Viterbi decoder data flow](image)

**Branch Metrics Calculations**

The hard and the soft decision decoders are differ in calculating the branch matrices'. In the hard decision decoder, a hamming distance will be obtained between the ideal pairs and the received pairs. Therefore each pair has a branch metrics. A branch metrics is measured using Euclidean distance. Let x defines the first received bit in the pair, y the second, Hence the branch metrics will be

\[ M_b = (x-x_0)^2 + (y-y_0)^2 \]

\[ M_b = (x_0^2-2xx_0+x_0^2) + (y_0^2-2yy_0+y_0^2) \]

\[ M_b^* = M_b - x^2-y^2 = (x_0^2-2xx_0) + (y_0^2-2yy_0) \]

The soft decision the actual results are not be needed as known absolute metric values. Only they differ in making sense. In the next steps the branch metrics are to be calculated without hard ware multiplications.

*Need to be calculated without the hard ware implementations. The obtained x and y values are to be computed with the 2’ complement.

**Path metric Calculation**

Path metric can be done by using ACS (Add-Compare-Select).this consists of repetition of each encoder state.
1. Add-out put response can be obtained by using the previous results with matching path values.
2. Compare, select-in this state system consists two path values, in that greater branch metric value should be leaved.

There exists $2(-1)$ survivor paths for $2(-1)$ encoder states the maximum difference cannot reach $\log(-1)$, where $d$ represents the difference between minimum and maximum branch metrics

Trace back:

It has been said that all paths combined after decoding a large block of data as shown in the figure. That is they only differ in their endings and have same beginning.

![Figure 6: An example for Survivor paths](image)

In figure 6 the Blue circles represents the encoder states. It can be seen that all survivor paths have a common beginning (red) and there will differ only in their endings. If we want over decoder to have final latency, we have to decode a continuous stream of data. The decoding bits related to this part can be sent to output, as it is common that some of the part at the starting of the graph that belongs to all survivor path. From the above statement the decoding can be performed as follows:

1. Finding the survivor path for N+D input bits.
2. Send $N$ bits to the output.
3. Find the survivor paths for another $N$ pairs of input bits.
4. Trace back from the end of any survivor paths to the beginning.
5. Go to step 2.

In this D parameter is important parameter and is called as decoding path. By increasing the D parameter there is a chance of decreasing of decoding error but also an increase in the latency. The maximum clock frequency is obtained for UMTS /GPRS after substitution and routing is equal to 32.26MHz. With out using the trace-back method achieved faster implementation of Viterbi decoder. The main purpose of using Viterbi decoder is for fast switching between UMTS and GPRS decoding which is used in software radio applications. area can be reduced by reusing the resources.

V. RESULTS AND DISCUSSIONS

The input values for Viterbi encoder are specified in the developed code. Providing input every time for the developed design is time taking process. So, the input values are included directly in the developed code, so that simulation process can be executed directly. The Viterbi encoder input values given in this project are:

```
0110100110010110100101100110100110010110011010
0101101001100101101001011001101001011010011001
0110011010
```

Each input value will be processed and a corresponding output will be provided for the Viterbi encoder. The output values will be specified in the form of wave forms in both Xilinx and active HDL simulation environments.

**A. Analysis of simulation using XILINX**

When clock signal is applied to the Viterbi decoder reset button is set as 0, and the system reset button kept as 0, the cyclic encoded data starts after 100ns by applying the valid encoded pulse train to the decoder enc_symbol 0 and enc_symbol 1 varies according to the periodic pulse train. According to the enc_symbol the encoded output bit also a pulse train.

The dec_symbol0 consists an 3’h0 error bits at decoder process up to 100ns, after 100ns it decodes 3bits at a time changes continuously every 50ns. dec_valid_in is a continuous pulse train when dec_symbol is change their bit stream every after 100ns.thus we get dec_out bit and dec_valid_out bits same as 1.the count of decoder is 103 and the buf_in_cnt as 102.from that the decoder buf_out_cnt and total count can be represented as 0 bit count.

The concerned output for the above encoder is

```
03121121102233221022131231100111022332231100
1121311202102233221022131231100
11213112021022332231100
```

---

Figure 7: Xilinx simulation environments for Viterbi encoder and decoder
B. Analysis of Simulation

For clock signal when set to value ‘1’ then it results in a continuous signal. When the signal is set to reset value ‘0’ then it is a dc signal, and there will be no changes in the signal. When srst that is system reset value is set to ‘0’. For an encoded bit input value is set to ‘0’ then the results can be obtained as 100110100111000110111000
When an encoded valid input value is set to ‘0’ then the result can be obtained as 101010101010101010101010
When an encoded symbol value is given as ‘0’ then the results are 110111000000001111111111
For an encoded symbol1 if the value is set to 1 then the analyzed results are 111011110001111000111111
Now for a decoded symbol0 when the value is set to 7 then there will be periodic changes obtained in a de-mux as 7 0 7 0 7
And for decoded symbol1 when the value is given as 7 then the output is the demux value given as 0 7 0 7 0 7
When decoded valid input value is given as ‘0’ then the output is continuous signal.
For a pattern value when given as ‘3’ then there will be no changes in the signal and similar appears when the decoded bit output, decoded valid output and decoded output error value is set to ‘0’.
When glb_seed value is set to 000E4048 then there appear a demux in the result. For ccnt, value is set to 1 then the result 104101010101010101010101010101010
And finally for buffer output count, total count and sim done value when set to ‘0’ there will be no changes in the signal.

C. Synthesis and Implementation

The design can be synthesized and implemented after verifying the design behavior with simulations. The process of synthesis and implementation can provide a clear view on how the developed code is working for obtaining simulation results.

The entire code developed for this project can be visualized using Xilinx hardware device. How the Viterbi algorithm code for encoder and decoder is working in a step by step process can be analyzed based on results of implementation.

Checking the details of the design after implementing each and every action is necessary. Synthesis report can provide an opportunity to view all the resources utilized by the developed design.
Implementing the design in FPGA editor can provide the below results. This view is providing in-built working process and routing information.

A user can verify the design before using it in a device; floor planner view can provide the opportunity to identify the Viterbi decoder position. Positions of all components used while developing the design can be obtained by just a click.

VI. CONCLUSION & FUTURE SCOPE

The Viterbi algorithm for the implementation of the convolution codes it concerns maximum memory, computational resource and makes utilization of power. The implementation part of Viterbi decoder includes the simulation, pipelining, decoding and the interleaving procedures. As Viterbi algorithm is conceived more interesting and challenging for this project topic, it is considered, and also it has wide variety of applications in digital communications field. This project helps to generate more profits by the developers using Viterbi algorithm. Anyone reading this document will have to gain the cognition of working with different tools like Xilinx ISE and Modelsim. Viterbi algorithm is successfully implemented using Verilog HDL hardware and tools like Xilinx and FPGA.

To attain the outturn of various hundred Mega Bits per second Viterbi algorithm is recommended to solve the problem of supplying power in case of applications which require high decoding throughput. By using a new coming called as relaxed Viterbi algorithm, the silicon area occupation and power consumption can be overcome, which provides even more better silicon area reduction and power saving. A less memory Viterbi algorithm is recommended as the Adaptive Viterbi algorithm requires very large amount of logic and memory for performing the functions. By using FPGA device and hybrid microprocessor the decoding benefits can be achieved in future. In future to improve the decoder performance the adaptive Viterbi algorithm is to carry out in reconfigurable hardware. The non binary codes can be implemented in the future for the Viterbi decoder. Viterbi decoder is now being implemented in Xilinx in future it can also be implemented using java. Therefore in the future Viterbi algorithm may be used for various scenarios. So in the future the complexity can be greatly reduced. By using M-algorithm decoding noise effects can also be greatly reduced.

VII. REFERENCES