Implementation of Viterbi algorithm using VHDL coding

Table of Contents

Chapter – 1: Introduction

1. 1. Overview

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 corruptions 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 [1]. 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. Al most 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 [2].Hardware description language called Verilog HDL is used to valuate this project, where it is one of the hardware descriptive languages that stand for Verilog Hardware Description Language. This language is employed in designing the electronic systems to semiconductor and electronic design industries as well as for assuring the analog and mixed signal circuit. This research makes use of two main tools namely MODELSIM – Simulation and XILINX-ISE – Synthesis for successfully reaching its objectives. Further of this research provides a clear description on Adaptive Viterbi Algorithm, its execution process and various kinds of languages and tools for evaluating the Viterbi algorithm.

1.2. Aim and Objectives:

Aim: Execution of Viterbi algorithm applying VHDL coding.

Objectives:

- To clearly understand the Hidden Markov model and Viterbi Decoder.

- To evaluate the basic functionalities and steps involved in Viterbi algorithm
- To research on the implementation of Viterbi algorithm through VHDL code
- To critically analyze the results obtained through VHDL code.

1.3. Purpose of Study

The main purpose of this study is to yield the gains obtained by the developers with the usage of viterbi algorithm. This research mainly centers on the grandness of Viterbi algorithm in the practical applications with the VHDL code. This research 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, VHDL code is used in order to implement the Viterbi algorithm in a proper way. Apart from various codes, researcher selected VHDL code for this research as it offers the high capability in designing the electronic systems. Apart from students and the business people, one can easily understand and analyze the Viterbi algorithm concepts and can gain more knowledge on the VHDL code and the tools that are used in this research.

1.4. Research Questions

- What are the basic functionalities that are used for designing the Viterbi algorithm?
- In what way Hidden Markov models are useful in implementing the Viterbi algorithm?
- Why viterbi algorithm is preferred for reducing the errors in the communication process?
- How one can implement the Viterbi algorithm through VHDL coding?

1.5. Research Context

This research usually concerned with the execution of Viterbi algorithm applying VHDL coding. The outlook of the project such as importance, developments, and applications will be examined by the researcher. The research 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 [3]. 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. Viterbi algorithm is advantageous since, it can achieve low power consumption and mainly error correction using Verilog HDL. VHDL includes Non-proprietary language, widely defended hardware description language (HDL), programming language, simulation language, Documentation language, functional verification language and usable for logic synthesis [4]. The researcher 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. [5]. with this research, 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 research to create the new algorithm which provides better features than the Viterbi algorithm. Hence, from this context it can be expressed that this research 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.

1.6. Research Method

Research is a probe of new facts that are exercised by the researchers. Generally, research method is an organized engineered which will determine the problems, suggest solutions and finally prepares the gathered data. For, research the data need to be gathered from many resources where the researcher will identifies proves to be collected and the techniques that need to applied in the research [6]. Generally there exists two methods for gathering the accurate data to the research. They are primary type and secondary type. In the primary type the researcher need to gather the data manually without referring or taking the ideas from other researchers where as in the secondary type the researcher gathers the data from numerous resources by referring the journals, magazines, books, etc [7]. For the present research, it's better to prefer the secondary resources as this research deals with the implementation of viterbi algorithm. Here, the researcher cannot depend only on primary data as the researcher will not find data by interviewing or surveying the people as all the people cannot know about this algorithm. So, it's better to prefer the secondary resources where the researcher can easily analyze the about the viterbi algorithm by referring to various journals and at last this algorithm can be implemented with the VHDL code to obtain the required result. In this research the researcher will go for a method of implementing VHDL programming to derive few areas of research objectives and also ensure that some areas are of research objectives will be derived and observed using the secondary data. At the same time the author will draw the conclusions based on the secondary data collected and the primary data gained from the experimental coding using VHDL programming language. At the same time the research questions are designed based on secondary data available in the initial research conducted by the researcher.

Chapter – 2: Literature Review

2.1. Overview

Virtebi 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. Virtebi 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 a numerable statements [8]. 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 thirds 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 rightest for a given HMM [9]. 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 needs 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.

2.2. 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 single bit errors can be adjusted by interpreting the real bit sequence during communication 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 1) Encryption: mainly used for security of data, 2) data compression: used for data streaming and to reduce the space, 3) Data translation: used to demonstrate the data for transmission of communication channels and 5) 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 [9]

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.

2.3. 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 [10].

Let the Hidden Markov Model(HMM) with the states may be Y, at initial stage probabilities π_i of being in state i and transition probabilities $a_{i,j}$ of transitioning from state i to state j. Say we observe outputs $x_0, \ldots, x_T$. The state sequence $y_0, \ldots, y_T$ most likely to have produced the observations is given by the recurrence relations.

$$V_{o,k} = P\ (x_o/k).\pi k$$

$$Vt_{,k} = P\ (x_t/k).\max y \in Y(ay,kV_{t-1,y}$$

Here $V_{t,k}$ is the probability of the most probable state sequence responsible for the first t + 1observations (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 Ptr (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. Then:

$$y_t \quad = \text{arg maxy} \in \text{Y}(V_{T,y})$$

y_{t-1} =Ptr (y_t,t)

The complexity of this algorithm is O ($T*IYI^2$).

2.4. Hidden Markov model and Viterbi decoder

2.4.1. 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 experimental 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 Markov model (HMM) is a signal facilitates to communicate with speech signals which achieved acceptance from almost all the communication systems. The fully discrete model with an idea of conditional independence had introduced the hidden Markov modes as a bivariate process. 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.

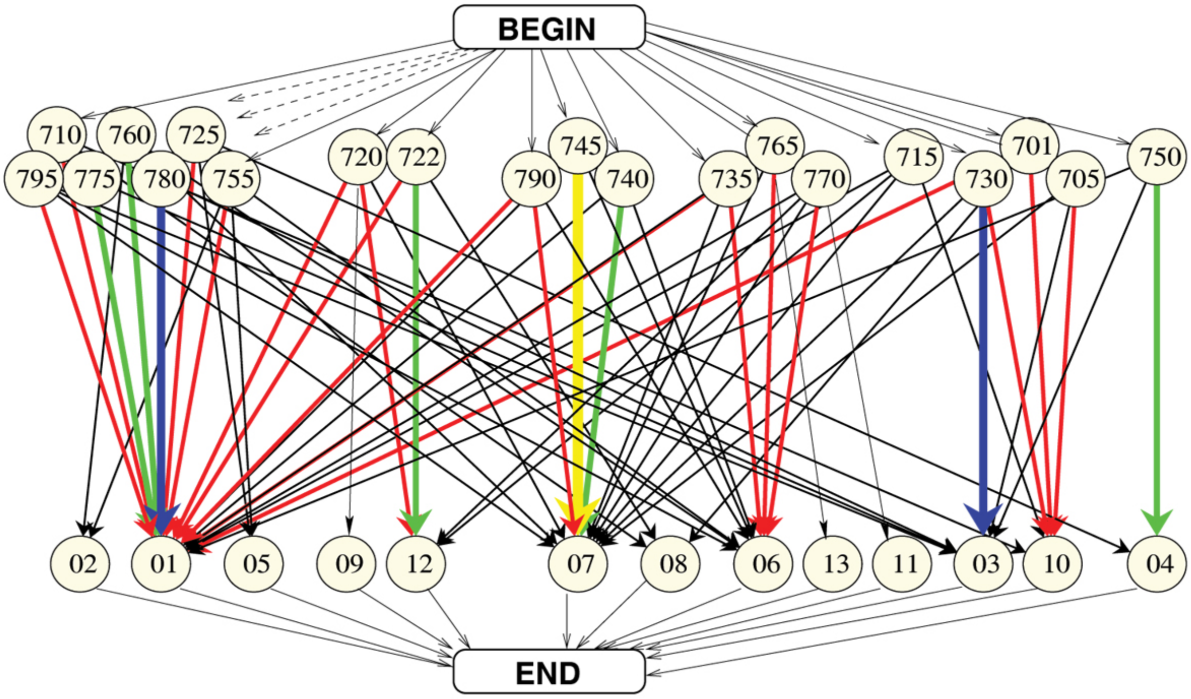


Figure 2.1: Shows the hidden Markov model [11]

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 there colors and shades are indicated as per the estimated probability counts based on training sequence.

2.4.2. 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 [12]:

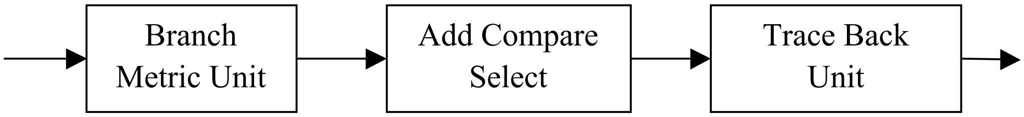
- Branch metric generation
- State metric generation
- Chain back

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:

In most of the Industry standard k = 7. Where (G0, G1) = (133, 171), rated at ½ Viterbi decoder. It is possible to implement both with Xilinx FPGA or ASIC. There are 256 latency clock cycle, Speed of the design is very high which is approximately up to 122 Mbps for the Virtex II at the same time for Spartan III the data rate is nearly 108 Mbps and more high for ASIC. The software input is of almost 4 bits. The length of track back will be of 64. Simple clock designs are completely synchronous.

2.5. 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. Kia, J. (2005, p.1) explains Viterbi algorithm as a "dynamic algorithm that uses certain path metrics to compute the most likely path of a transmitted sequence" [13]. 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.



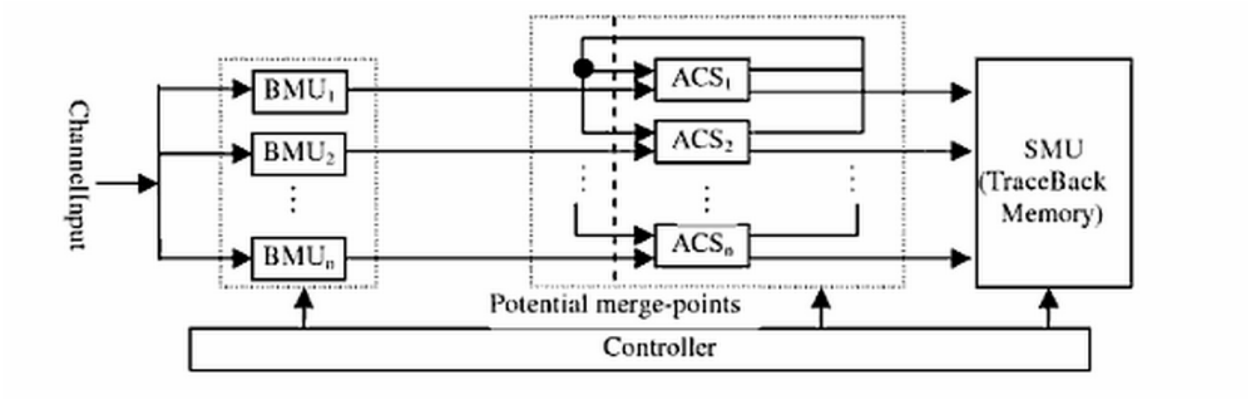


Figure – 2.2: Shows the basic block diagram of Viterbi decoder [14].

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.

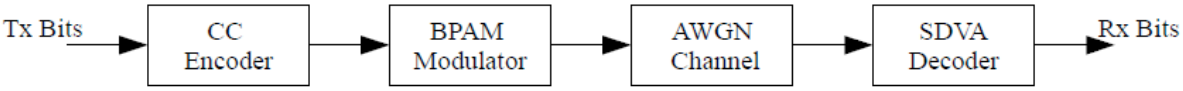


Figure – 2.3: Shows the block diagram of Viterbi algorithm [15].

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.

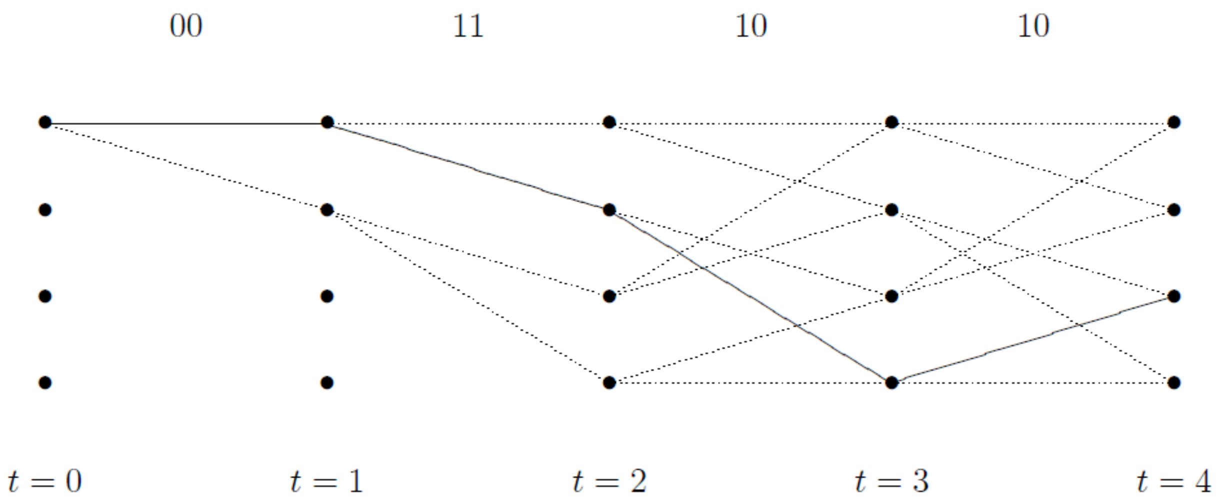


Figure 2.4: Shows the Viterbi algorithm trellis [15].

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

$$BM\ (rr, bb) = (r_0\text{-}b_0)^2 + (r_1 + b_1)^2$$

= r02−2r0b0+b02+r12−2r1b1+b12

= 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 = - b_0 as well as r_1 = - b_1 and BM= -2

The path metric (λ) in the minimum Euclidean distance in the trellis does not required the actual value the original order of the floating point pair numbers is

λnew=λprev+r0b0+r1b1

The path metric λ 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 [15].

2.6. 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.

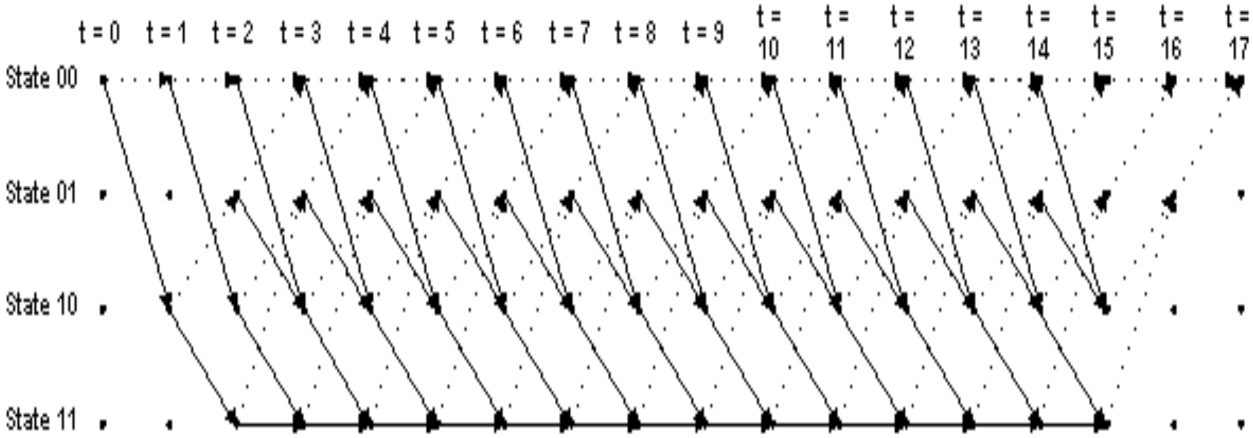


Figure 2.5: shows the trellis diagram of viterbi algorithm. [16]

18 columns shows the time instants from t0 to t17 both t=o 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.

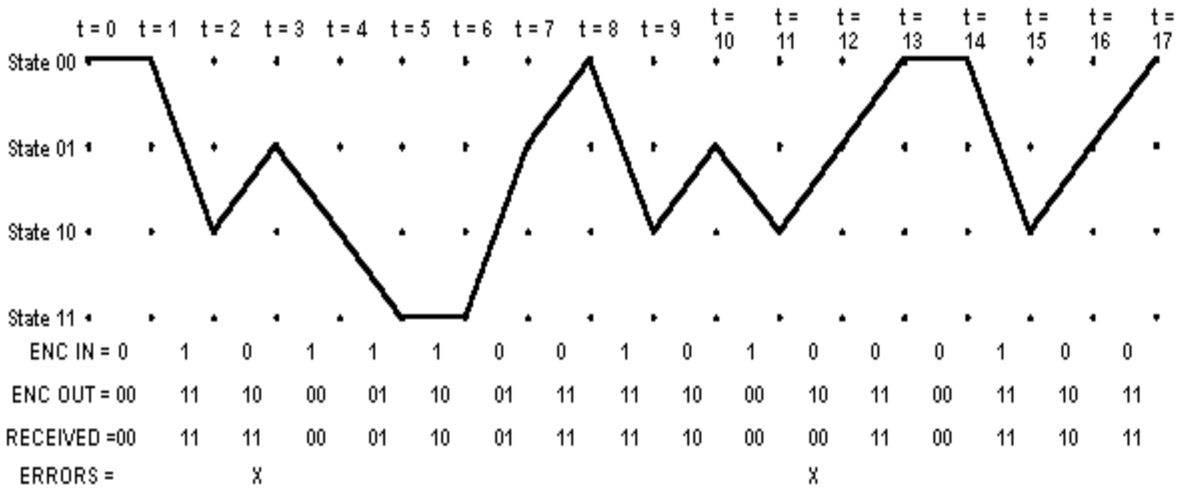

t = 0
t = 1
t = 2
t = 3
t = 4
t = 5
t = 6
t = 7
t = 8
t = 9
t = 10
t = 11
t = 12
t = 13
t = 14
t = 15
t = 16
t = 17
State 00
State 01
State 10
State 11
ENC IN = 0 1 0 1 1 1 0 0 1 0 1 0 0 0 1 0 0
ENC OUT = 00 11 10 00 01 10 01 11 11 10 00 10 11 00 11 10 11
RECEIVED =00 11 11 00 01 10 01 11 11 10 00 00 11 00 11 10 11
ERRORS = X X

Figure 2.6: shows states of the trellis diagram of viterbi algorithm. [16]

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.

2.7. 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 [16].

2.8. Summary

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.

Chapter 3: Research on Proposed solution

3.1. Overview

Research is generally defined as the human activity that is carried out based on the intellectual application in the investigation of matter. Basically there are various approaches for the researcher to finish their task successfully but these people select the approaches depending on their research objectives [17]. For the present research, it's better to prefer the secondary resources when compare to the primary resources. As the researcher may face problems while gathering the accurate

3.2. Language used for Viterbi algorithm

There are number of functional programming languages, since most of the hardware programs are written in hardware description language such as VHDL (Very High Speed Integrated Circuits) hardware description language which may not be programmed through imperative languages like C or MATLAB. 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 [18]. 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. 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. Let us consider a simple example for this type. Let a car has three states. They are forward stop reverse. And a transition is made that reverse to forward is not allowed [19]. To verify the Viterbi algorithm MATLAB code should be short. TMS320C54x DSP assembly language is used to write the Viterbi decoding algorithm. In this report a detailed execution of Viterbi algorithm is provided.

The Viterbi coder you will implement 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. This allows us to use the specialized instruction set supported by the C54x DSP's

3.3. Importance of VHDL over other languages

VHDL is known to be the standards of Very High Speed Integrated Circuit (VHSIC) as it is a hardware description language. VLSIC is the advanced technology in designing the new technique VLSI silicon chips, which was proposed in 1980s by United State government as well as air force. VHDL describes the behavior for the electronic circuit/ system. The VHDL is the earliest hardware descriptive language which was originally accepted by the Institute of electrical and electronics engineers of IEEE 1076 standards as well as later IEEE 1164 is an additional standard introduced for multi-valued logic system [20]. The designing process of the VHDL synthesis involves a cycle of six steps which are illustrated as follows: [21]

- Analysis on all the requirements of design and define them: The design requirements are for providing the essential setup, clock requirements, and maximum operating frequency in the critical paths of the project.
- Design the VHDL code along with its description: basically a design methodology is made use of for describing any system some of the common design methodology are Top-down, bottom-up and flat
- The source code of the process is to be simulated: As the simulation would be done before synthesis and design stage it is most essential feature of hardware and description languages.
- Achieve a synthesized, best optimized and to fit the design: creation of design descriptions and logic circuits are done by the designers in the synthesis tools of VHDL. Optimization process compresses the circuit by reducing the amount of product terms. Placing the logical products in al logic device is called fitting process.
- Next the design is to be simulated again: the design element of VHDL includes entity declaration and body of architecture.
- Finally the processed design is to be implemented.

From the above steps it can be concluded that the circuit synthesis and circuit simulation are the main task to be carried out by VHDL. The synthesis circuit in PLD or FPGA or ASIC are the major utilities of VHDL. The designer starts writing the code with an extension of .VHDL while saving the ENTITY's same name by compilation which is the conversion of high-level VHDL language as shown in the figure below. The next level of performance is on the gate-level net-list for rapid operation at a register transfer level (RTL) for simulation. The place and route filter software will generate practical layout for PLD chip and generate masks for ASIC.

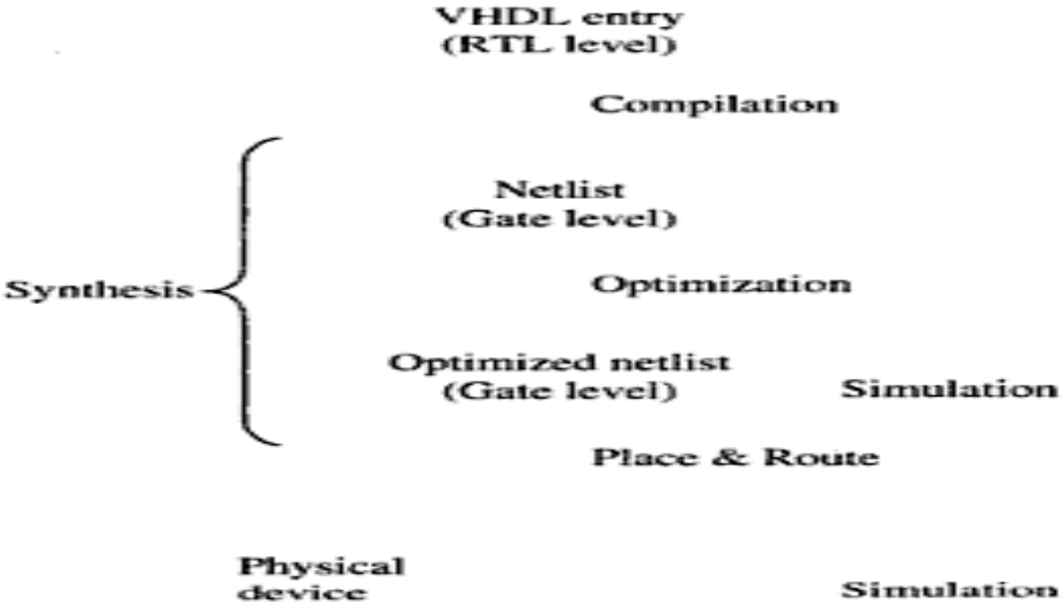


Figure 3.1: shows the design flow of VHDL [21]

Design process includes analysis, elaboration, simulation and synthesis. During analysis verify the syntax and semantic errors in the design. Identify all the steps which are involved in the design unit separation and which are placed in the library. Flatten the design chain of command and finalize the results of elaboration of collected signal process. Simulation of the executed module and use discrete events for sensitive input signals. Simulation algorithm will initializes all the phases of transactions. The translated register-transfer-level (RTL) to gate-level-netlist is functioned in the synthesis process. The electronic components execution in many areas is described by VHDL. VHDL simulation model is created by the accurate view of electric circuit behavior. VHDL simulation model can be used as a basis for constructing big circuits [21]. VHDL functions as a universal programming language. VHDL resembles C and C++ languages. VHDL allows concurrent issues description that provides a strong set of control and data representation characteristics which make VHDL different from other languages. VHDL is mainly used to point the function of a circuit. Text models are normally written by using VHDL which depicts a logic circuit that is refined by a synthesis program. The logic design is tested by using a simulation program. The design is interfaced by the logic circuits using the simulation models. The collection of all these simulation models is called as testbench. The parallelism integral in hardware designs is managed by VHDL [21]. But the parallel constructs differs from the syntax. VHDL is a universal language and has filed input and output potentialities. Executable binaries can also be chassis by some VHDL compilers. Then VHDL is used to write a test bench to assert the functionality of design using files on the host computer so that to define stimuli and the results are compared with the user. To produce a successful simulation it may be easy for an inexperienced developer, but to synthesis into a real device it cannot be done and even it is too large. To produce a RTL schematic of a wanted circuit one has to design hardware in a VHDL IDE. The waveforms of inputs and outputs of the circuit after getting appropriate test bench are showed by verifying the generated schematic. The inputs have to be determined correctly to bring forth an appropriate testbench for a particular circuit.

Advantages: Before the synthesis tools transform the design into real hardware VHDL is used for system design and grants the behavior of the required system to be modeled and simulated which is the key advantage of VHDL. VHDL is a dataflow language and allows a description of concurrent system which is another advantage of VHDL. Synthesizable constructs and VHDL templates [22].Simulation of electronic and synthesis of such designs are two different goals often used in VHDL. FPGA or an ASIC is an execution technology in which VHDL is compiled and mapped. This process is called as synthesis. To synthesize VHDL many FPGA sellers have free tools for employ with their chips, whereas ASCI tools are of very high cost. For synthesis all constructs in VHDL are not worthy. Different capacities are associated with various syntheses. For language constructs map into similar hardware for all synthesis tools there exists a common synthesizable subset. The official synthesis subset that is considered as a subset of the language is defined by IEEE 1076.6. To write very idiomatical code for synthesis as results it is considered as a best practice. VHDL is the IEEE standard technology independent language as it is the vendor portable and reusable. VHDL is mostly used in the fields of application specific integrated circuits and programmable logic devices but, it is not a programming language. Some of the basic functional areas of the VHDL are [23]:

- VHDL components facilitates for using an array as well as regular analysis of its structure.
- All the component algorithm of the VHDL describes hook-up.
- Some of the other components like N-bit array memory, J-bit memory, K-register file etc are also used frequently.

As the VHDL language code is processed it may be used in two purpose one for implementation and programmability of the device like Altera, Xilinx, Atmel etc in the process of fabrication to ASIC chip. And the second is to approach for number of complex designs for the process of commercial chips, for the entity, architecture, package and control.

3.4. Tools used along with the VHDL

The two main tools which are used in the Modelsim SE/DE and Xiliniex ISE which are described as follows: Modelsim SE and Modelsim DE are the two basic commercial tools available. Modelsim SE uses UNIX (Sun Solaris, IBM AIX, HP-UX), Linux (Redhat 6.0 - 7.2), Microsoft Windows (98, 2000, NT, XP) operating system. Modelsim DE uses 32-bit Windows XP/Vista and Linux platforms. Both of these are available in 5.6 and 5.5 versions respectively. [24] The combination of unified debug environment for Verilog with single kernel simulator (SKS) technology, VHDL, and System C was first done by mentor graphics. For both FPGA and ASIC design, MODELSIM became a choice for simulation by the compounding of industry-leading, native SKS performance. In almost all tools flows process it is the best standard and platform.

ModelSim PE - Simulation and Debug: Modelsim PE Features: Modelsim PE is the windows based simulator for VHDL, Verilog or mixed language simulation environments, and is leading in the industry. Features of Modelsim PE are described in further sections. It has partial VHDL 2008 support. It has windows 7, Secure IP support. The transaction wlf logging support in all languages including VHDL It has system C option. It has RTL and Gate-Level Simulation. VHDL and System Verilog Design, Integrated Debug, Verilog, , Mixed-HDL Simulation option, Code Coverage option, Windows 32-bit, Enhanced debug option are some of the features of Modelsim PE.[24]

Modelsim PE Benefits: Integrated project management simplifies managing project data ,Easy to use with outstanding technical support ,Intuitive GUI for efficient interactive debug, Awardwinning technical support are the benefits of ModelSim PE,Sign-off support for popular ASIC libraries, Cost effective HDL simulation solution Upgradeable to ModelSim SE for higher performance, ASIC sign-off ,.

Overview of Modelsim PE: ModelSim PE is entrance level simulator, and it offers Verilog, VHDL. ModelSim is known for its high execution delivering, simplicity in use, and outstanding product support. It alters transparent mixing of Verilog and VHDL in one design. The outstanding performance of native compiled code which is done by the platform independent compile is allowed by the ModelSim architecture. Verilog and VHDL language standards are fully supported by ModelSim PE [25]

ModelSim DE - Simulation and Verification: The choice of simulator for all leading electronics companies is ModelSim DE

ModelSim DE Features: easy-to-use GUI with TCl interface, Intelligent, System C option available, Code coverage, Standard support for Xilinx Secure IP, source code templates, integrated project management, Native compiled, Single Kernel Simulator technology are some of the features of ModelSim DE.

ModelSim DE Overview: ModelSim technology enables the transparent mixing of Verilog and VHDL languages in one design. The design quality can be improved by assertion based verification. The cause of failure can be easily examined by the simulation results in only the wave window caused when complex assertions are triggered. A powerful simulation can be delivered by ModelSim DE. It will be well suited for small and medium sized FPGAs verifications, especially designs with critical functionality and that are complex. [26]

XILINX ISE is Xilinx Integrated Software Environment. FPGA and CPLD designs are designed in XILINX. World class design entry, synthesis and executiontools are offered by ISE. Various versions are available for installing ISE. Various design steps can be brought up by ISE as it is a graphical User Interface. This can be used in markets like communications, industrial, consumer, automotive and data processing. In design flow the user creates the source files based on design objectives using any HDL like Verilog, VHDL etc. After design entry and simulation the project synthesis is done. For HDL devices and its design process it can be said that XILINX ISE is a dominant software tool.

3.5. Building Viterbi algorithm

The coding in Viterbi Algorithm was introduced in the year 1967 for the first time in one of the literature paper [26, P.1]. Since then, due to its easiness in implementation, it has been used to many different areas related to decoding problems. VHDL is a hardware description language. For the use in design process systems this has become a practical tool because this application is used to define the operation than the logic of the design which leads to the diminution of costs and evolution time. Recently VHDL code can be mapped on FPGAs. This method is available to the designer which reduces the costs of ASIC designs. As Digital signal processing is now used in many areas of human life. The demand for more complex ASICs is gradually increasing. Like Viterbi algorithm and matched filters many important algorithms used in digital communication systems contains the calculation of square of some sample value.

Usually the maximum-likelihood path given the input sequence is reconstructed by the Viterbi algorithm. Let us define some of the following terms: [27]

A soft decision decoder: A reliable estimation is made on a decoder receiving bits from the channel. To complete a task three bits are normally sufficient. If the width of the soft decision is increased further then there will be only slight increase in the performance but there will be a significant increase in the computational difficulty.

A hard decision decoder: It is a decoder which receives only bits without any reliable estimate from the channel,

A branch metric: it is the difference between the ideal pairs and the received pairs.

A path metric: A path metric is the sum of the all branches in the path.

The path with the minimal metric path is the maximum-likelihood path. Therefore the decoding problem is almost equal to the problem of finding such a minimal path.

Let us suppose assume that we know a minimal path metric ending for almost every possible encoder in this state. There will be only two states for an encoder to move to that state in any given encoder and branch metrics must be known for both of these transitions. Therefore in the next step, in any given state there will be only two paths ending. If the path has lesser metric then the path is called as survivor path and the other path is dropped less likely. Thus a minimum metric path can be known on next step, and can repeat the above procedure.

For the Viterbi algorithm in general the fundamental process is assumed to be Markov process with some characteristics is: Finite state, in this the number M is finite. Discrete time, this means that same unit of time is taken from going to one state of position to another state of position. The sequence of observations depends probabilistically only on the previous sequence transitions which is observed in memory less noise.

3.6. Building of VHDL

VHDL (VHSIC hardware description language) is a programming language. The behavior of field programmable gate arrays can be illustrated by this language. The electronic components execution in many areas is described by VHDL. VHDL simulation model is created by the accurate view of electric circuit behavior. VHDL simulation model can be used as a basis for constructing big circuits. VHDL functions as a universal programming language. VHDL resembles C and C++ languages. VHDL allows concurrent issues description that provides a strong set of control and data representation characteristics which make VHDL different from other languages. VHDL is mainly used to point the function of a circuit. Text models are normally written by using VHDL which depicts a logic circuit that is refined by a synthesis program. The logic design is tested by using a simulation program. The design is interfaced by the logic circuits using the simulation models. The collection of all these simulation models is called as test bench. The parallelism integral in hardware designs is managed by VHDL. But the parallel constructs differs from the syntax. VHDL is a universal language and has file input and output potentialities. Executable binaries can also be chassis by some VHDL compilers. Then VHDL is used to write a test bench to assert the functionality of design using files on the host computer so that to define stimuli and the results are compared with the user. To produce a successful simulation it may be light for an inexperienced coding engineer, but to synthesis into a real device it cannot be done and even it is too large.

To produce a RTL schematic of a wanted circuit one has to design hardware in a VHDL IDE. The waveforms of inputs and outputs of the circuit after getting appropriate test bench are showed by verifying the generated schematic. The inputs have to be determined correctly to bring forth a required test bench for a particular circuit

3.7. Synthesizable constructs and VHDL templates

Simulation of electronic designs and synthesis of such designs is two different goals often used in VHDL. FPGA or an ASIC is an execution technology in which VHDL is compiled and mapped. This process is called as synthesis. To synthesize VHDL many FPGA sellers have free tools for use with their chips, whereas ASCI tools are of very high cost.

For synthesis all constructs in VHDL are not worthy. Different capacities are associated with various syntheses. For language constructs and idioms map into common hardware for many synthesis tools there exists a common synthesizable subset. The official synthesis subset that is considered as a subset of the language is defined by IEEE 1076.6. To write very idiomatic code for synthesis as results it is considered as a best practice.

3.8. How Viterbi algorithm is implemented through VHDL

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 Viterbi decoder also used to acquire exact expectation of transmitted sequence passed through a medium with inter symbol interference (ISI).In order to decode conventional codes Viterbi maximum likelihood algorithm is the effective computational procedure. Especially for limited length conventional codes the Viterbi decoder computes very accurately. It is also decodes conventional codes at high speed. The graphical representation of Viterbi algorithm is known as trellis diagram. 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.

List all the array states gritty with trace back and named it as state sequence array.

3.8.1. The Viterbi Decoder Architecture

The Viterbi decoder is seperated 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 is 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[27]. 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 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.

3.8.2. The 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]. By using the Field programmable gate arrays (FPGA) there is break time in the design of telecommunication devices that are based on the requirements that customize and recycle the Verilog HDL. Field programmable gate arrays are the semiconductor devices that maintain programmable logics and inter connectors. All the systems will be implemented in the Verilog (HDL).To build the RTL level VHDL for decoder the synthesis tool is used. Initial conditions that are placement of the weights and the paths need to modify the state are readily coded in the VHDL. Field programmable gate arrays (FPGA) are the extremely worth full tools in the design of the VLSI. Users determine the FPGA with the hardware description language (HDL). 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 advanced FPGA technologies and the well developed electronic design automatic tools (EDS) have made it possible to realize the Viterbi decoder. The main motivation of this is to use the VHDL synthesis and the simulation tools to realize the Viterbi algorithm. 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 .The decoder implemented in VHDL can be used to generate in MAT LAB for the noisy, encoded streams of data which were to be quantized using the common bit width [29]. 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. Then at last the decoder is synthesized and simulated on by writing the VHDL description.

FPGA is used to implement Viterbi algorithm. The design is described by the hardware description language VHDL. The implementation of this design is done on Xilinx Spartan 2e xc2s100-5cpq208. Using Xilinx tool Viterbi decoder is first synthesized and simulated for the proposed pipeline structure. First let us have a brief introduction about Xilinx software and VHDL. Traceback method (TB) and register exchange method are the two techniques used for decoding the data. XILINX: Xilinx is capable of planning; formulating and markets programmable logic services by devices those changes day by day with integrated circuits. Software design tools, determined system derivatives delivered as intellectual property (IP) cores, design services, customer training, field engineering and proficient support. Xilinx sells both FPGAs and CPLDs programmable logic devices for electronic equipment makers in end markets such as communications, industrial, consumer, automotive and data processing. The emphasis is on writing solid synthesizable code and enough simulation code to write a viable test bench. Structural, Register Transfer Level (RTL), and behavioral coding styles are covered [30].

3.9. Summary

A convolutional code that uses the maximum-likelihood method was decoded by the Viterbi algorithm procedure. Formal definition of algorithm, the algorithm may be resumed formally as:

For each i,, i = 1, ... , n, let :

$$\mathbf{X}_i = (X_{i_1}, X_{i_2}, \ldots, X_{i_T})$$

This names the probability calculations by gathering the product of the initial hidden state probabilities with the related observed probabilities. [31]

The errors found in the convolution coded sequence are decoded basically by using Viterbi algorithm. To find the coded sequence based on the transmission signals trellis diagram is used by the Viterbi algorithm. Trellis diagram is the common supporter to study the Viterbi algorithm.

C or MATLAB are the languages used for Viterbi algorithm. Two Viterbi algorithms, namely isolated sign language Viterbi algorithm and continuous sign language Viterbi algorithm are used to search the frame at all the same time. The sequence of hidden states that were found is called as Viterbi path the Viterbi algorithm is used which is a dynamic programming algorithm. To verify the Viterbi algorithm MATLAB code should be short. TMS320C54x DSP assembly language is used to write the Viterbi decoding algorithm.

VHDL is known to be the standards of Very High Speed Integrated Circuit (VHSIC). The behavior of field programmable gate arrays can be illustrated by this language. VHDL functions as a universal programming language. VHDL resembles C and C++ languages. VHDL is mainly used to point the function of a circuit. In VHDL the test benches were used to assert the functionality of design with the usage of files on the host computer so that to define stimuli and the results are compared with the user.

Before the synthesis tools transform the design to real hardware VHDL is used for system design and grants the behavior of the required system to be modeled and simulated which is the key advantage of VHDL. VHDL is preferred apart from other languages because it is a independent technology and moreover it is a IEEE standard technology, but it is not a programming language. There are no tools to generate VHDL test benches automatically.

Chapter 4: Analysis on Research

4.1. Overview

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.

4.2. 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 in 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.

For better use of power and coding rates in circuits' adaptive channel coding systems are used. The adaptive decoder algorithm must contain programmability or re-programmability at hardware logic level or algorithmic level in order to increase the computation levels. Field programmable gate array device is mentioned here for achieving both parameters power and coding rate.

The fundamental concept used in Software radio terminal is re-programmability which is present either at hardware level or algorithmic level. An interface with different communication standards are used in Software defined radio terminals and it must consist of high decoding capability for reprogrammable devices. Viterbi decoder use convolution codes for channel decoding in its programmable architecture and is similar to a special class of standards and specifications.

Viterbi decoder program is used for switching between UMTS and GPRS .whenever the implementation uses only decoder, reconfiguration is not a major problem. FPGA holds the total transceiver digital processing sub-system in real time applications. In order to decrease the problems in design process re use the same building blocks. At transmission time for particular application programmability is used where data coding rate is continuously changed at run time.

The Viterbi decoding process conventional codes of fixed size of 5 and 9 and the code rate ½ and 1/3.by using FPGA there is possibility of implementing extra features in the Viterbi decoder (e.g. different standards and new functions).

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 [32]

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.

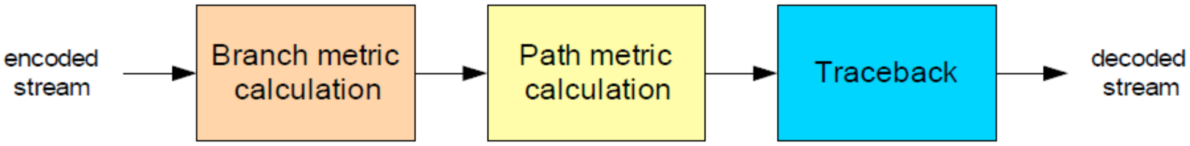


Figure 4.1: shows the Viterbi decoder data flow [35]

IMPLEMENTATION:

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

$$M_b=(x-x_0)^2+(y-y_0)^2$$

$$M_b=(x^2-2xx_0+x_0^2)+(y^2-2yy_0+y_0^2)\ ;$$

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

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`complemant.

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 vales.

2. Compare, select-in this state system consist two path values, in that greater branch metric value should be leaved.

Their exists 2(□−1) survivor paths for 2(□−1) encoder states.the maximum difference cannot reach

□log□−1,Where δ 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.

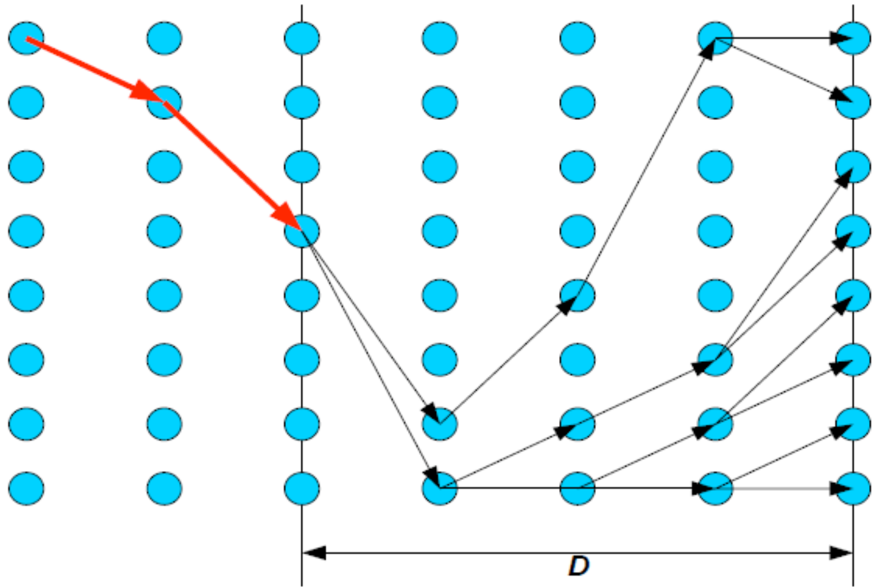


Figure 4.2 shows is an example for Survivor paths [35]

In the graph of 4.2 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: [33]

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. Traceback 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.

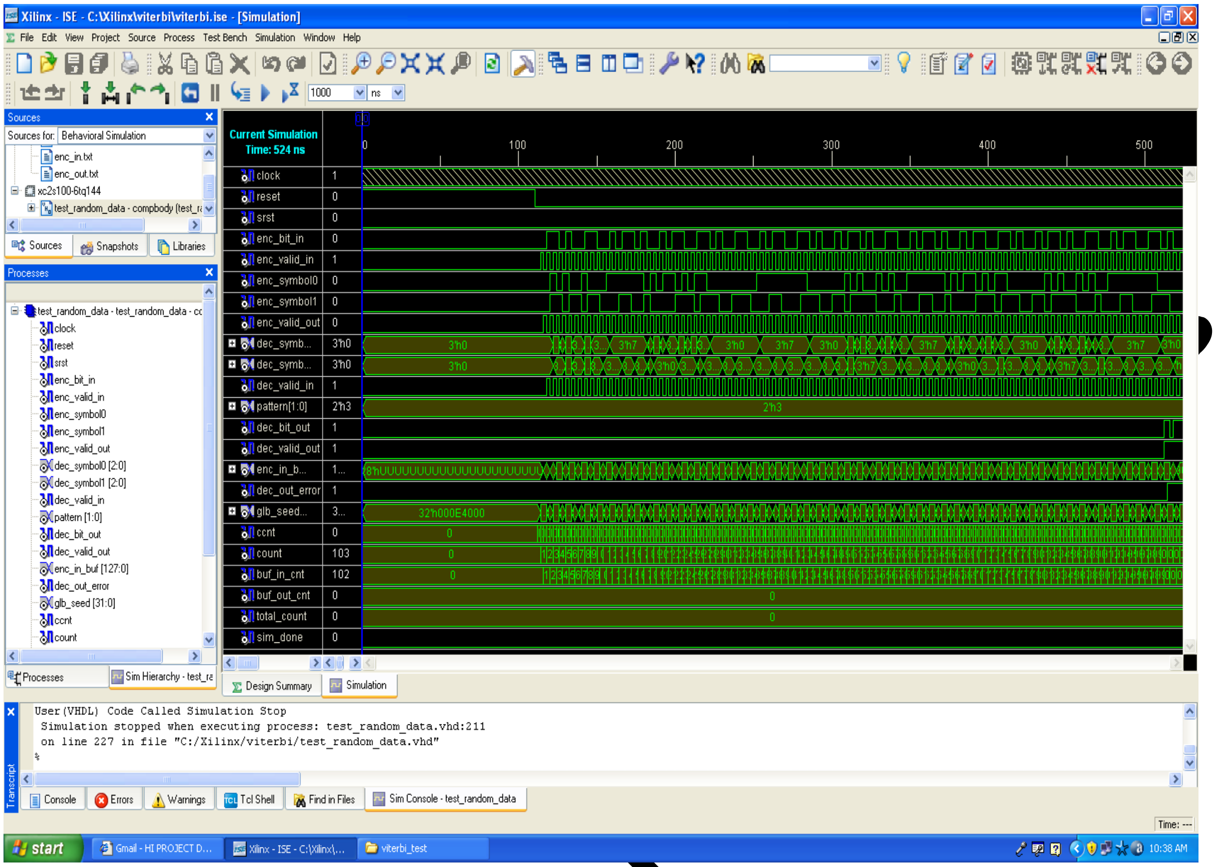
Summary: The VD has been tested with a Nallatech extreme DSP Development Kit [37], and the appropriate results can be compared with both functional and MATLAB simulations. The maximum clock frequency is obtained for UMTS /GPRS after substitution and routing is equal to 32.26MHz.with out using the trace-back method achieved faster implementation of Viterbi decoder. The extra resources used for reconfiguration procedure compared to fixed implementation is just 5%.this result is satisfied with procedures in Ref [17]. 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.

4.3 Results

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:

01101001100101101001011001101001100101100110100101101001100101101001011001101 0
01011010011001011001101 0

Each input value will be processed and 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.



```
User(VHDL) Code Called Simulation Stop
 Simulation stopped when executing process: test_random_data.vhd:211
 on line 227 in file "C:/Xilinx/viterbi/test_random_data.vhd"
%
```

Figure 4.1: Xilinx simulation environments for Viterbi encoder and decoder

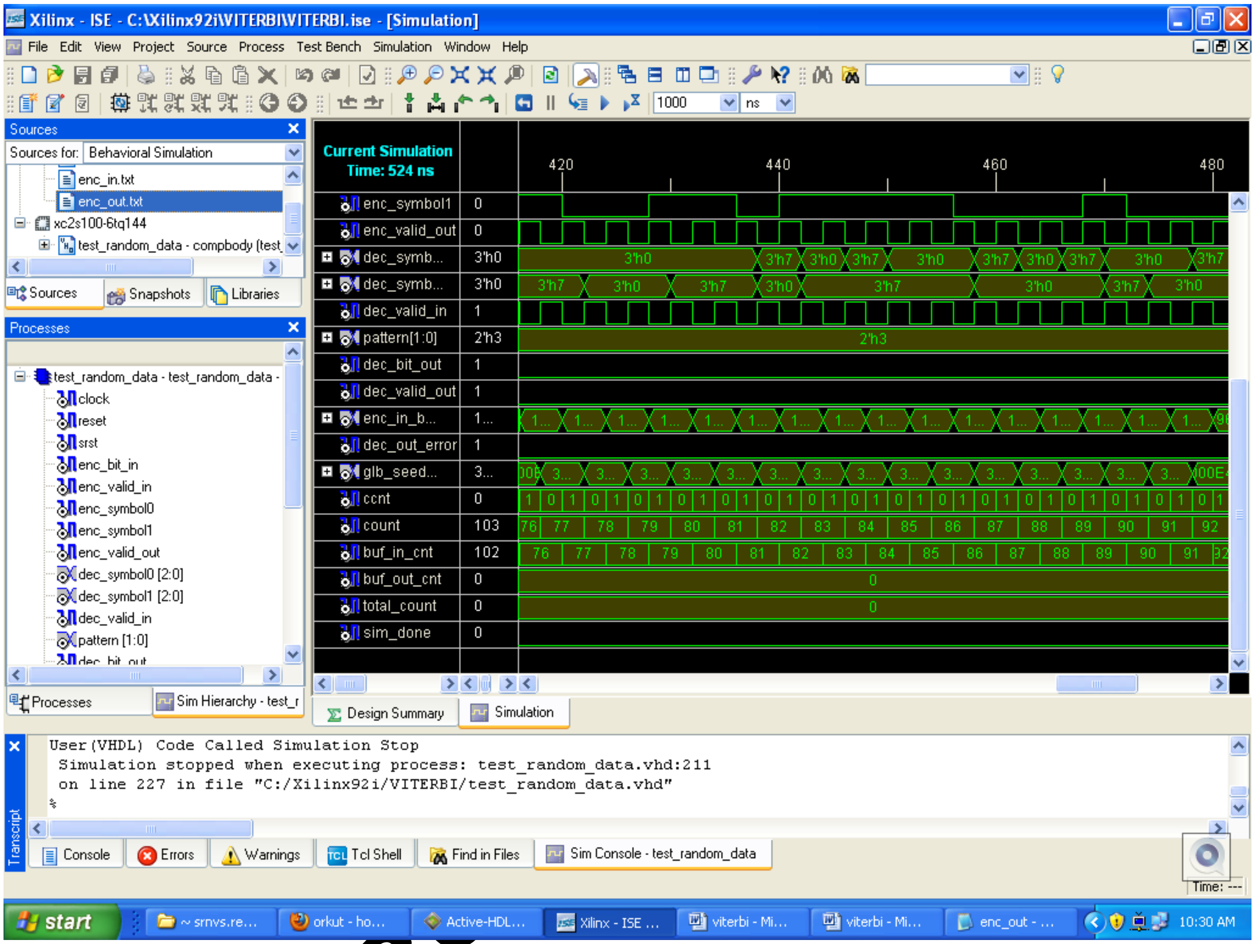
ANALYSIS OF SIMULATION USING XILINX

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

031211211022332212022131231100111022332223110011213112021022332212022131231100 112131120210223322231100



```
User(VHDL) Code Called Simulation Stop
 Simulation stopped when executing process: test_random_data.vhd:211
 on line 227 in file "C:/Xilinx92i/VITERBI/test_random_data.vhd"
%
```

Figure 4.2 Xilinx simulation results for Viterbi encoder

Analysis of simulation using Active VHDL

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

1010101010101010101010

When an encoded symbo0 value is given as ‘0’ then the results are

1101110000000111111111

For an encoded symbol1 if the value is set to 1 then the analyzed results are

1110111100011110001111111

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 0 7

When decoded valid input value is given as ‘0’ then the output is continues 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’.

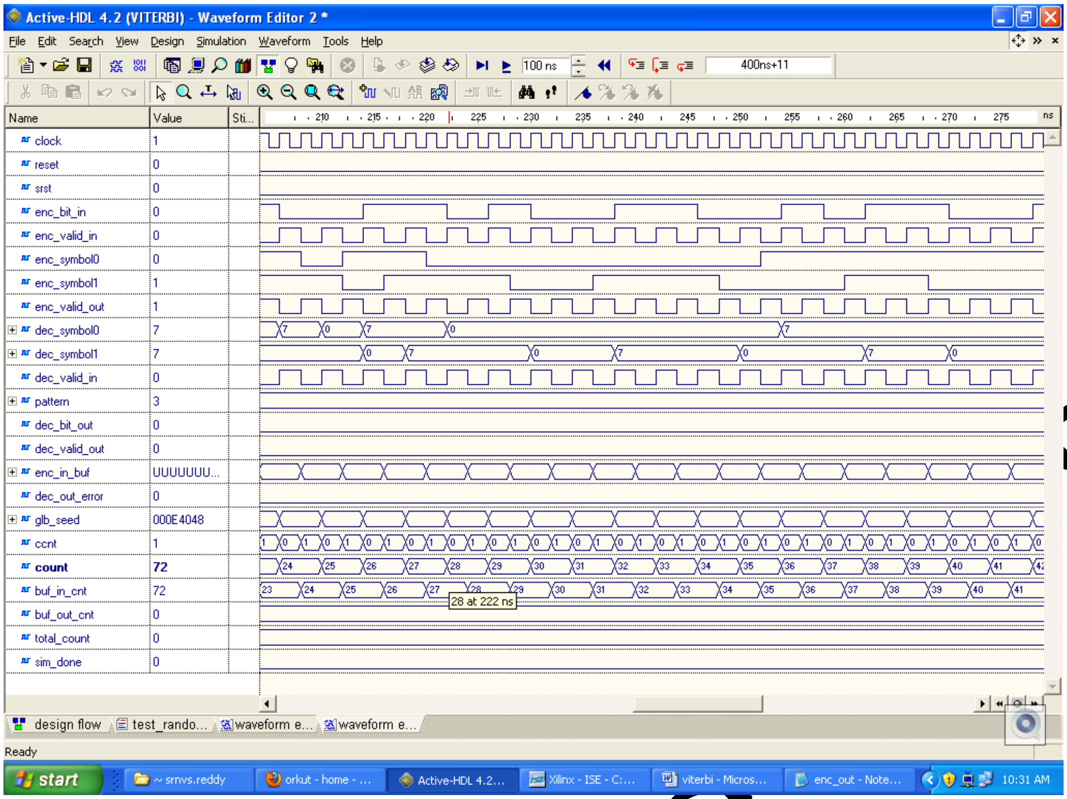
When glb_seed value is set to 000E4048 then there appear a demux in the result. For ccnt, value is set to 1 then the result

104101010101010101010101010101010

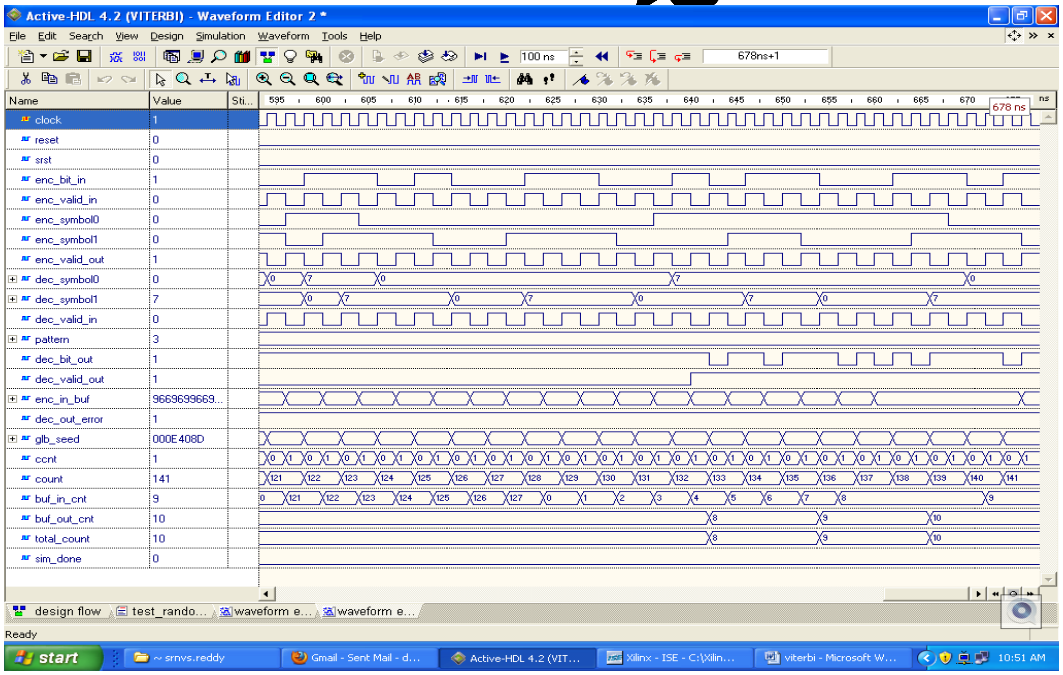
And finally for buffer output count, total count and sim done value when set to ‘0’ the n there will be no changes in the signal.

Implementation of Viterbi encoder can also be simulated with the help of active HDL simulation environments. Active-HDL is an Aldec product developed using Field Programmable Gate Array (FPGA) with HDL simulator. The main reason for choosing this tool for simulation is that it can support graphical and text based designs with various simulation languages.

The below figures can represent the active-HDL simulation results.

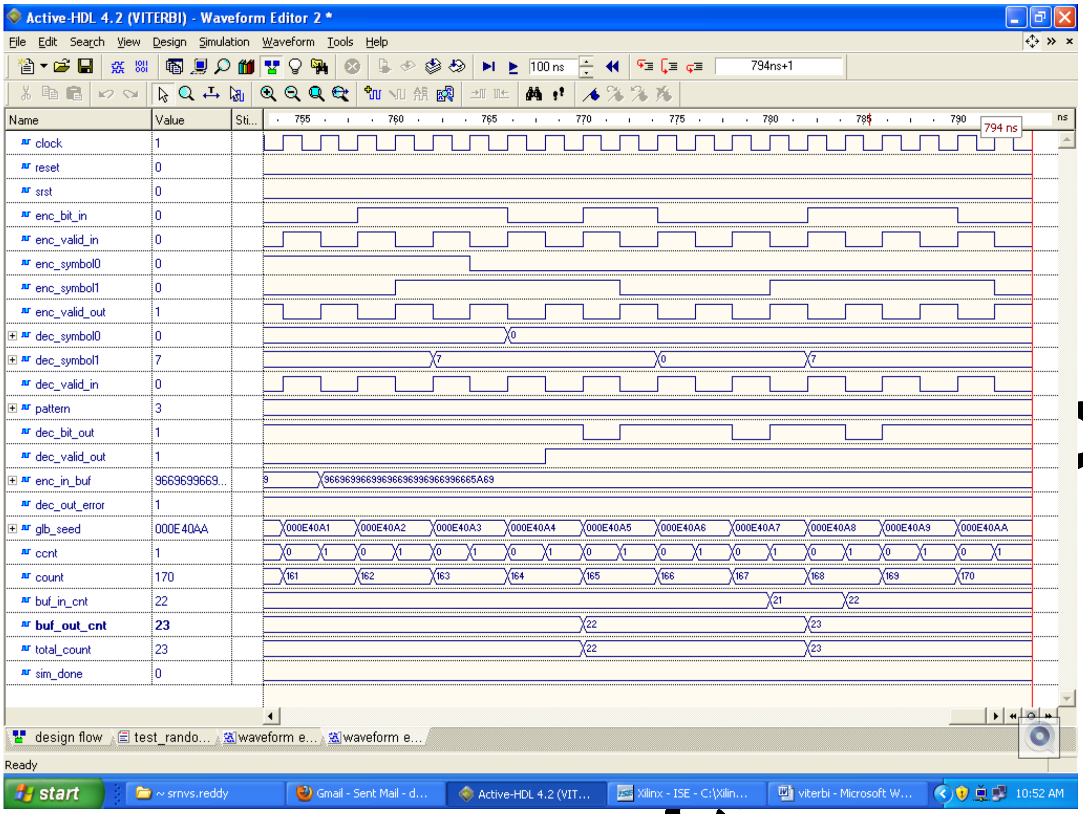


Name	Value
clock	1
reset	0
srst	0
enc_bit_in	0
enc_valid_in	0
enc_symbol0	0
enc_symbol1	1
enc_valid_out	1
dec_symbol0	7
dec_symbol1	7
dec_valid_in	0
pattern	3
dec_bit_out	0
dec_valid_out	0
enc_in_buf	UUUUUUU...
dec_out_error	0
glb_seed	000E4048
ccnt	1
count	72
buf_in_cnt	72
buf_out_cnt	0
total_count	0
sim_done	0

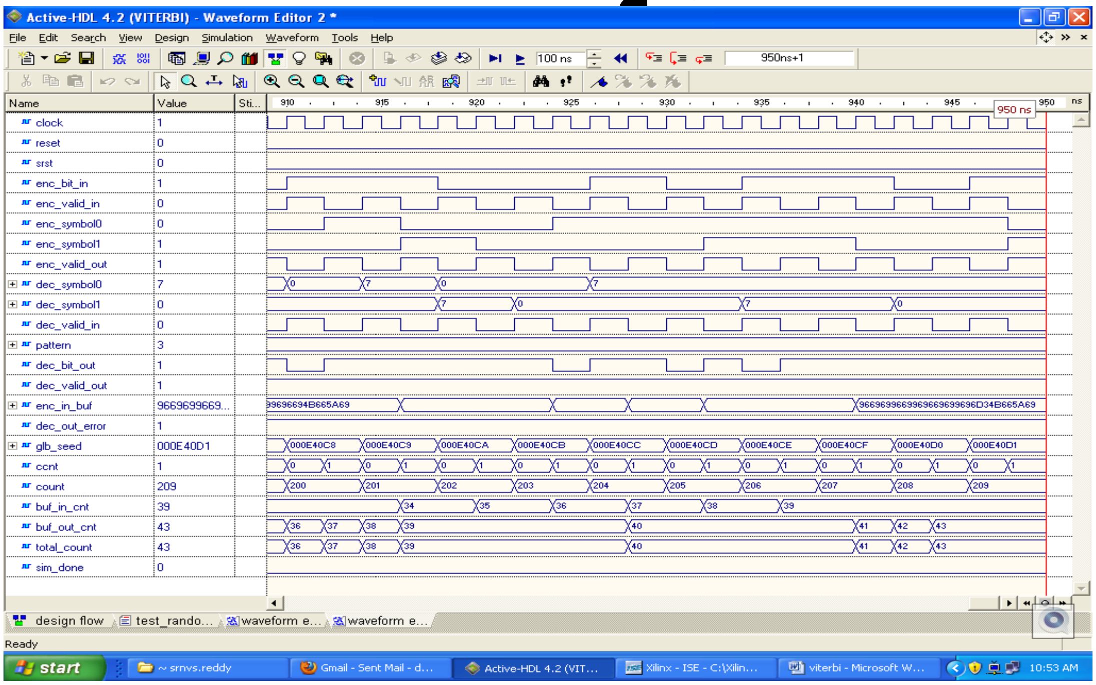


Name	Value
clock	1
reset	0
srst	0
enc_bit_in	1
enc_valid_in	0
enc_symbol0	0
enc_symbol1	0
enc_valid_out	1
dec_symbol0	0
dec_symbol1	7
dec_valid_in	0
pattern	3
dec_bit_out	1
dec_valid_out	1
enc_in_buf	9669699669...
dec_out_error	1
glb_seed	000E408D
ccnt	1
count	141
buf_in_cnt	9
buf_out_cnt	10
total_count	10
sim_done	0

Figure 4.3: Simulation using active-HDL



Name	Value
clock	1
reset	0
srst	0
enc_bit_in	0
enc_valid_in	0
enc_symbol0	0
enc_symbol1	0
enc_valid_out	1
dec_symbol0	0
dec_symbol1	7
dec_valid_in	0
pattern	3
dec_bit_out	1
dec_valid_out	1
enc_in_buf	9669699669...
dec_out_error	1
glb_seed	000E40AA
ccnt	1
count	170
buf_in_cnt	22
buf_out_cnt	23
total_count	23
sim_done	0



Name	Value
clock	1
reset	0
srst	0
enc_bit_in	1
enc_valid_in	0
enc_symbol0	0
enc_symbol1	1
enc_valid_out	1
dec_symbol0	7
dec_symbol1	0
dec_valid_in	0
pattern	3
dec_bit_out	1
dec_valid_out	1
enc_in_buf	9669699669...
dec_out_error	1
glb_seed	000E40D1
ccnt	1
count	209
buf_in_cnt	39
buf_out_cnt	43
total_count	43
sim_done	0

Figure 4.4: Viterbi encoder simulation using active-HDL

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

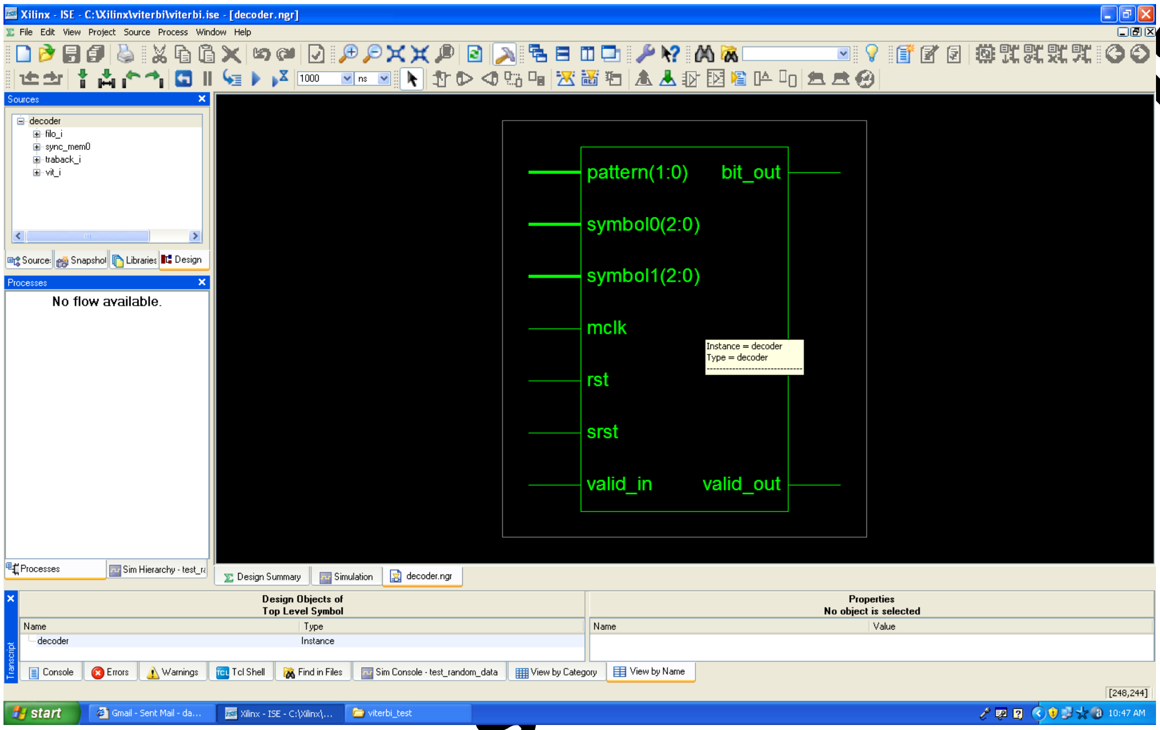
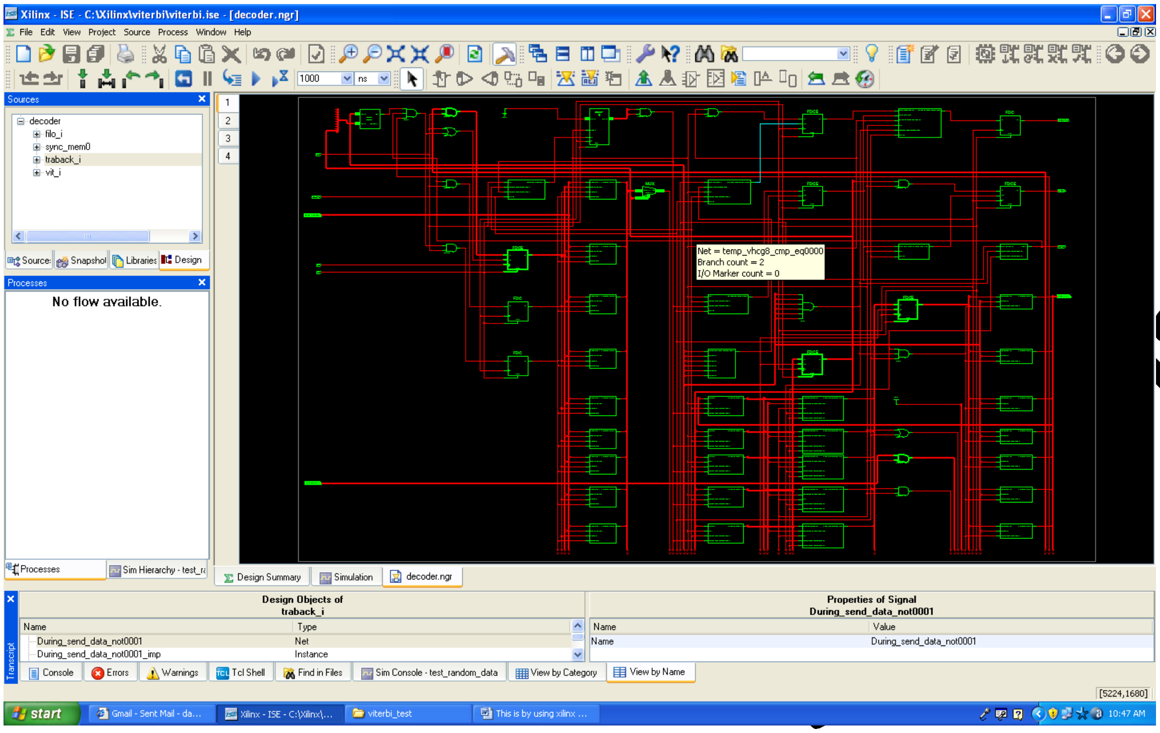


Figure 4.5: Viterbi decoder synthesis

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.



No flow available.

Net = temp_vhcg8_cmp_eq0000
Branch count = 2
I/O Marker count = 0

Design Objects of traback_i

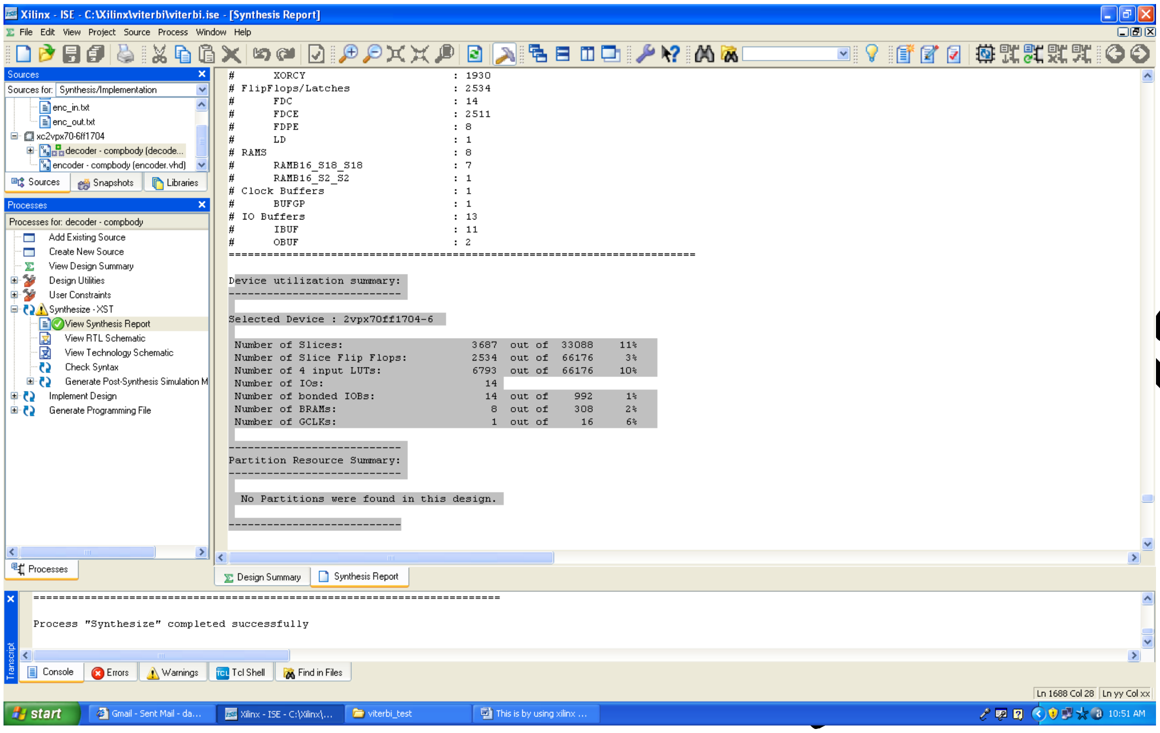
Name	Type
During_send_data_not0001	Net
During_send_data_not0001_imp	Instance

Properties of Signal During_send_data_not0001

Name	Value
Name	During_send_data_not0001

Figure 4.6: Design layout for the developed code using Xilinx-ISE

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

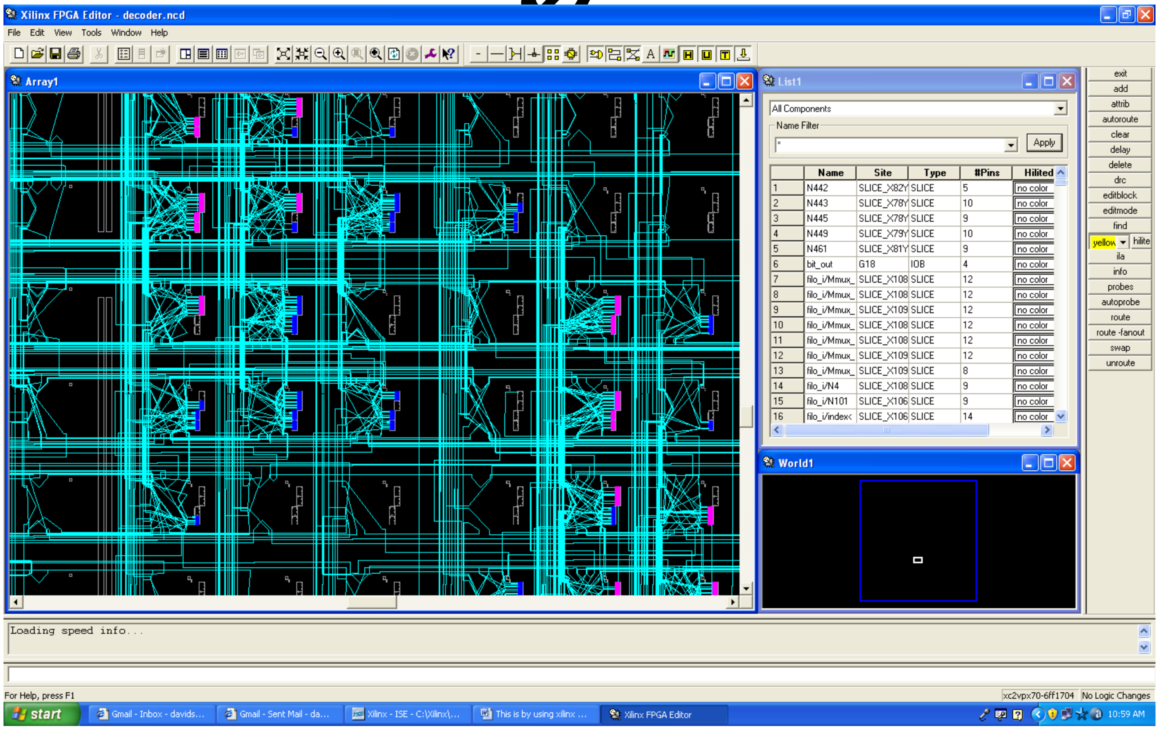


```
#        XORCY                          : 1930
# FlipFlops/Latches                     : 2534
#        FDC                            : 14
#        FDCE                           : 2511
#        FDPE                           : 8
#        LD                             : 1
# RAMS                                  : 8
#        RAMB16_S18_S18                 : 7
#        RAMB16_S2_S2                   : 1
# Clock Buffers                         : 1
#        BUFGP                          : 1
# IO Buffers                            : 13
#        IBUF                           : 11
#        OBUF                           : 2
=========================================================================

Device utilization summary:
---------------------------

Selected Device : 2vpx70ff1704-6

 Number of Slices:                    3687  out of  33088    11%
 Number of Slice Flip Flops:          2534  out of  66176     3%
 Number of 4 input LUTs:              6793  out of  66176    10%
 Number of IOs:                         14
 Number of bonded IOBs:                 14  out of    992     1%
 Number of BRAMs:                        8  out of    308     2%
 Number of GCLKs:                        1  out of     16     6%

---------------------------
Partition Resource Summary:
---------------------------

  No Partitions were found in this design.

---------------------------
```

```
=========================================================================

Process "Synthesize" completed successfully
```

Figure 4.7: synthesis report of the design

Implementing the design in FPGA editor can provide the below results. This view is providing in-built working process and routing information.



All Components

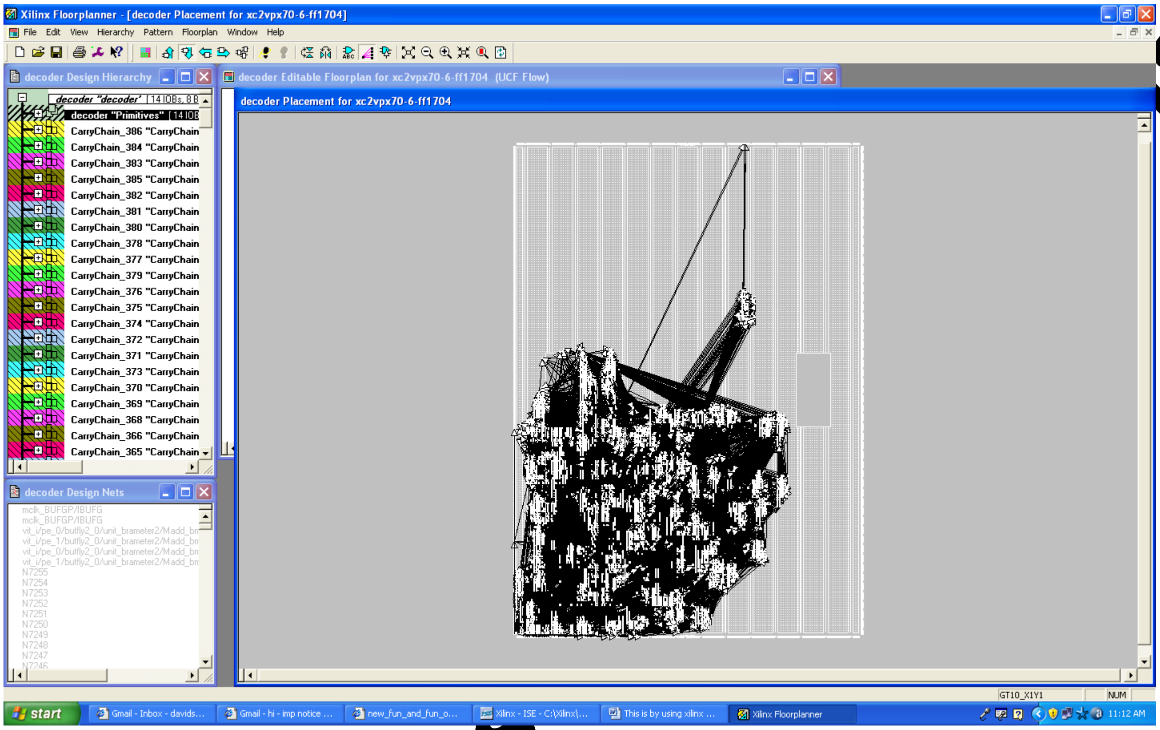
Name Filter: *

	Name	Site	Type	#Pins	Hilited
1	N442	SLICE_X82Y	SLICE	5	no color
2	N443	SLICE_X78Y	SLICE	10	no color
3	N445	SLICE_X78Y	SLICE	9	no color
4	N449	SLICE_X79Y	SLICE	10	no color
5	N461	SLICE_X81Y	SLICE	9	no color
6	bit_out	G18	IOB	4	no color
7	filo_i/Mmux_	SLICE_X108	SLICE	12	no color
8	filo_i/Mmux_	SLICE_X108	SLICE	12	no color
9	filo_i/Mmux_	SLICE_X109	SLICE	12	no color
10	filo_i/Mmux_	SLICE_X108	SLICE	12	no color
11	filo_i/Mmux_	SLICE_X108	SLICE	12	no color
12	filo_i/Mmux_	SLICE_X109	SLICE	12	no color
13	filo_i/Mmux_	SLICE_X109	SLICE	8	no color
14	filo_i/N4	SLICE_X108	SLICE	9	no color
15	filo_i/N101	SLICE_X106	SLICE	9	no color
16	filo_i/index<	SLICE_X106	SLICE	14	no color

```
Loading speed info...
```

Figure 4.8: Design and implementation of Viterbi decoder

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.


Xilinx Floorplanner - [decoder Placement for xc2vpx70-6-ff1704]
File
Edit
View
Hierarchy
Pattern
Floorplan
Window
Help
decoder Design Hierarchy
decoder Editable Floorplan for xc2vpx70-6-ff1704 (UCF Flow)
decoder Placement for xc2vpx70-6-ff1704
decoder Design Nets
GT10_X1Y1
NUM
start
Gmail - Inbox - davids...
Gmail - hi - imp notice ...
new_fun_and_fun_o...
Xilinx - ISE - C:\Xilinx\...
This is by using xilinx ...
Xilinx Floorplanner
11:12 AM

decoder "decoder" [14 IOBs, 8 B
decoder "Primitives" [14 IOB
CarryChain_386 "CarryChain
CarryChain_384 "CarryChain
CarryChain_383 "CarryChain
CarryChain_385 "CarryChain
CarryChain_382 "CarryChain
CarryChain_381 "CarryChain
CarryChain_380 "CarryChain
CarryChain_378 "CarryChain
CarryChain_377 "CarryChain
CarryChain_379 "CarryChain
CarryChain_376 "CarryChain
CarryChain_375 "CarryChain
CarryChain_374 "CarryChain
CarryChain_372 "CarryChain
CarryChain_371 "CarryChain
CarryChain_373 "CarryChain
CarryChain_370 "CarryChain
CarryChain_369 "CarryChain
CarryChain_368 "CarryChain
CarryChain_366 "CarryChain
CarryChain_365 "CarryChain

mclk_BUFGP/IBUFG
mclk_BUFGP/IBUFG
vit_i/pe_0/butfly2_0/unit_brameter2/Madd_br
vit_i/pe_1/butfly2_0/unit_brameter2/Madd_br
vit_i/pe_0/butfly2_0/unit_brameter2/Madd_br
vit_i/pe_1/butfly2_0/unit_brameter2/Madd_br
N7255
N7254
N7253
N7252
N7251
N7250
N7249
N7248
N7247
N7246

Figure 4.9: Floor planner view for the design

Hence from the above discussion on results of this project it can be concluded that, Viterbi encoder and decoder is implemented by developing VHDL coding and the developed code is simulated using Xilinx and active-HDL tools. During the implementation process, a user can obtain all the details related to the design such as component positions, process of working and resource utilization, if the code is executed in Xilinx tool.

Chapter – 5: DISCUSSION

5.1. Overview

The viterbi decoder as a convolution decoder maintains a powerful method in forward error correction. Cellular telephones with low data rates had a wide range of applications with viterbi decoder. They were used as the trills code for demodulation in telephones. It is the minimum decoding algorithm in solving the convolution codes for the transmission obtained at additive white Gaussian noise channel. The viterbi algorithm for the implementation of the convolution codes it concerns maximum memory, computational resource and makes utilization of power. Adaptive viterbi algorithm will be answered as a solution for this problem. When the HDL code is executed in the Xilinx simulation environments, then the basic specification will be displayed. The generation of all successive code words is concerned as the received word. The output errors in “burst” can`t be fixed by the viterbi decoder .The convolution codes are used as the generally in pair with Reed-Solomon codes. In the development process the HDL code the wave form that can be used for the simulation is created. Constrain file and the timing pin location will be developed in design. It is designed in a way where the user needs to provide the inputs every time. When the programmed is executed the simulation results are achieved. The viterbi decoder is specifically implementing the viterbi algorithm on the VLSI chips. Here it is the using in advanced techniques in viterbi decoder that can increase the throughput by multipath fading in Adaptive algorithm [37].

5.2. Suggestions

Field programmable gate arrays (FPGA) are the VLSI chips that can be performed high end applications as well as. As the FPGA`s are capable of performing the basic logic gates and other functions that provide routing. The programmable components maintain the decoders, flip-flops and other complex element blocks that operate on the specific function. The RTL level for decoding purpose can be performed by the VHDL with the FPGA kit. It benefits more with the usage of the FPGA kit. The implementation part of viterbi decoder includes the simulation, pipelining, decoding and the interleaving procedures. As it is suggested that the FPGA kits provide the high end operating Spartan 3 FPGA kit is one that offering the low cost and high performance and other applications. Xilinx is Integrated Software Environment (ISE) tool that can be provided for the implementation by addition of the pre defined functions. The placement, mapping and synthesis can be achieved by this. [38] The physical and the practical implementations of viterbi algorithm on a decoder afford the accurate results streams because of the quantization of the input signal, path metrics, branching and also the trace back length.

5.3. Comparisons

The implementations part can be achieved on various methods some of them can be mentioned as the Verilog HDL on the Xilinx tool, Java coding and C#. The verilog coding on the Xilinx tool provides more reliable result set. It can be also be upgraded as a future scope of the project. One way in the software implementations to achieve the ACS butterfly using the assembly language and allowing instruction set extensions to speed up decoding time is the most time investing pattern. Comparison between the one and the other designs the Spartan FPGA kit is the less price and flexible thing that provides a LCD in which the regular messages can be shown. Many electronic applications are used by it. [39]The adaptive viterbi decoder generally requires large amount of memory and the circuitry logic for performing the operations so for this reason as possible as less occupied viterbi algorithm architecture is required.

Chapter 6: Conclusions

As Viterbi algorithm is conceived more interesting and challenging for this research topic, it is considered, and also it has wide variety of applications in digital communications field. This research helps to generate more profits by the developers using Viterbi algorithm. Anyone besides students can easily analyze these Viterbi algorithm concepts and can gain more knowledge about it. This research mainly concerned with implementation of Viterbi algorithm using VHDL coding. Viterbi algorithm has many advantages like low power consumption and main advantage is error correcting using VHDL. Anyone reading this document will have to gain the cognition of working with different tools like Xilinx ISE and MODELSIM. [39]

The chance of getting errors is more often because communication is a process of transferring data from one point to other and it involves a lot of coding process. By interrupting the original bit sequence simple bit errors can be solved and by using some of the important features of Viterbi algorithm arbitrary problems can be solved randomly. Some of the general techniques for error correction are forward error correction (FEC), auto repeat request (ARQ), hybrid ARQ and error code correction (channel coding). C or MATLAB are the languages used for Viterbi algorithm. Two Viterbi algorithms, namely isolated sign language Viterbi algorithm and continuous sign language Viterbi algorithm are used to search the frame at all the same time. To find the sequence of hidden states which is called as Viterbi path the Viterbi algorithm is used which is a dynamic programming algorithm. To verify the Viterbi algorithm MATLAB code should be short. TMS320C54x DSP assembly language is used to write the Viterbi decoding algorithm.

VHDL is known to be the standards of Very High Speed Integrated Circuit (VHSIC). The behavior of field programmable gate arrays can be illustrated by this language. VHDL functions as a universal programming language. VHDL resembles C and C++ languages. VHDL is mainly used to point the function of a circuit. VHDL is used to write a test bench to assert the functionality of plan using files on the host system so that to define stimuli and the results are compared with the user.

The main advantage of Viterbi algorithm is the description will be low even in the presence of more errors and the algorithm works more effectively. Another advantage of using this Viterbi algorithm is due to its cost effectiveness [40].

For implanting this, tools used along with it are MODELSIM – Simulation and XILINX-ISE. ModelSim SE and ModelSim DE are the two basic commercial tools available. XILINX ISE is Xilinx Integrated Software Environment. Xilinx ISE is a predominant software tool for developing HDL devices and its design process. To reduce power consumption high speed applications and the gate level simulators of Viterbi decoder is used in these decoders. To get a proper state at time, Viterbi decoder is concerned with various processing elements. Two register files one write and read are used by the radix butterflies. The bits are stored using the path metric file. In multipath fading adaptive algorithm is used. The throughput can be increased with the usage of the advancement of techniques in Viterbi decoder. Finally, Viterbi algorithm is successfully implemented using Verilog HDL hardware and tools like Xilinx and FPGA. Results that were obtained are to be observed and the entire developed code working process, its design and synthesis results can be obtained very easily using Xilinx ISE and FPGA editors. The Adaptive Viterbi algorithm requires very higher memory locations and logical programming capable performance the operations, so there is a need to less occupied memory Viterbi algorithm architecture has to be developed. The algorithm parameters are needed to be chosen and also the required variations for the noise level predict better results [41].

6.1. Recommendations

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. FPGA kits are used in the research for the implementation. Large amount of time in milliseconds will be spend by FPGA which is used by the Adaptive computing to overwrite the data and extra power consumption for charging the assemble data. FPGA guides to temporary growth by this of it response time and can be fatal in the communication path [42]. Therefore dynamically reconfigurable Processors are used in order to overcome these dynamically reconfigurable devices. Low signal to noise ratio is observed by the problem of localization principle. 3-D Viterbi search is recommended to overcome this trouble. Advanced versions for the algorithm are recommended as Viterbi algorithm has high throughput.

6.2. Future scope

By using FPGA device and hybrid microprocessor the decoding benefits can be achieved in future. Power benefits are provided by the integration of sequential decoding. To reduce the multipath fading which damages the signals, the adaptive array technique is used for future satellite communication. The solutions of the Adaptive Viterbi decoding calculate on the chosen noise level and algorithmic parameters. For independence on noise level and fixed complexity M-algorithm is used. In future to improve the decoder performance the adaptive Viterbi algorithm is carried out in reconfigurable hardware. [43]For power saving techniques can be used for the power saving architecture can be designed for the above decoder which is executable in the mobile devices. 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.

Bibliography:

[1] R. E. Stake (1995) “The art of case study research”. USA: Sage.

[2] K.Aleksandar, M.F. Jose (2000) “The Viterbi Algorithm and Markov Noise Memory”. [Internet] available at URL: <http://www.ece.cmu.edu/~moura/papers/kavcic_viterbinoisememory.pdf>, [accessed on 20th November 2010].

[3] K. Hueske, J. Geldmacher, J. Gotze (2007) Adaptive decoding of convolutional codes, Copernicus Publications, pp. 209-214. Internet. (N.D) Coding in communication system. [Internet] available at URL: < http://www.mathdb.org/notes_download/elementary/number/ne_N.pdf>, [accessed on 20th November 2010].

[7] B. Tristan (2006) IMPLEMENTATION OF THE VITERBI ALGORITHM USING FUNCTIONAL PROGRAMMING LANGUAGES. [Internet] available at URL: <http://people.eecs.ku.edu/~esp/publications/c2009ItcTristan.pdf>, [accessed on 20th November 2010].

[8] K. Hucker (2001) Research Methods in Health, Care and Early Years. UK: Heinemann Publishers.

[9] Y. K., P. Cheung, G. A. Constantinides, J.T. D. Sousa (2003) Field-programmable logic and applications: 13th International Conference, FPL 2003, Lisbon, Portugal. New York: Springer.

[10] J. Kia (2005) What Is a Convolutional Encoder / Convolutional Encoding. [Internet] available at URL: <http://www.askkia.com/articles/what-is-a-convolutional-encoder-viterbi-algorithm.html>, [accessed on 20th November 2010].

[11] K.Aleksandar, M.F. Jose (2000) “The Viterbi Algorithm and Markov Noise Memory”. [Internet] available at URL:

<http://www.ece.cmu.edu/~moura/papers/kavcic_viterbinoisememory.pdf>, [accessed on 20th November 2010].

[12] R. E. Stake (1995) “The art of case study research”. USA: Sage.

[13] W. F. Lee (2000) VHDL coding and logic synthesis with Synopsys, USA: Academic Press.

[14] K. Yaser (n. d) Introduction to VHDL. [Internet] available at URL: <http://www.engr.newpaltz.edu/~khalify/VHDL.pdf>, [accessed on 20th November 2010].

[20] I. Filip, M. Katerina (2002) VHDL Toools [Internet] available at URL: <http://www.cesnet.cz/doc/techzpravy/2002/vhdltools/>, [accessed on 20th November 2010].

[21] Xilinx (2008) Synthesis and Simulation Design Guide, [internet] available at URL: <http://www.xilinx.com/itp/xilinx10/books/docs/sim/sim.pdf>, [accessed on 20th November 2010].

[22] A. P. Volnei (2004) Circuit design with VHDL. United States of America: MIT press.

[23] P. P. Chu (2006) RTL hardware design using VHDL: coding for efficiency, portability, and scalability. USA: John Wiley and Sons.

[24] Project seminar (2010) VHDL (VHSIC hardware description language) [Internet] available at URL: <http://www.seminarprojects.com/Thread-vhdl-vhsic-hardware-description-language--14251#top>, [accessed on 20th November 2010].

[33] Viterbi Algorithm (n.d.) hidden Markova models. [Internet] available at URL: <www.comp.leeds.ac.uk>, [accessed on 20th November 2010].

[34] Mentor Graphics (2008) “ModelSim Tutorial”, [internet] available at URL: < http://www.actel.com/documents/modelsim_tutorial_ug.pdf>, [accessed on 20th November 2010].

[35] CORE (2004) Viterbi Algorithm for Decoding of Convolutional Codes. [Internet] available at URL: < http://www.1-core.com/library/comm/viterbi/viterbi.pdf>, [accessed on 20th November 2010].

[36] E. Robert, P. R. Daniele (2009) “Optimizing the Viterbi Algorithm and Applications to Supervised Sequential Learning” [internet] available at URL: <http://jmlr.csail.mit.edu/papers/volume10/esposito09a/esposito09a.pdf>, [accessed on April10th 2010].