http://www.jctjournals.com ISSN(ONLINE):2278 – 3814

Volume-10, Issue-07,July, 2021
JOURNAL OF COMPUTING TECHNOLOGIES (JCT)
International Journal
Page Number: 01-08

Design And Implementation Of An Enhanced Viterbi Decoder Using VHDL With Improved Performance For SDN

Shashikant Kumar Singh[1], Prof. Manish Gupta[2]
[1]*M.Tech Scholar,* [2]*Assistant Professor*
[1,2]*Dept. of Electronics & Communication Engineering*
[1,2]*Scope Engineering College, Bhopal MP, India*
[1]*shashi.rgpv@gmail.com*

***Abstract*—In this research work discussed about design and implementation of an enhanced viterbi decoder using VHDL with improved performance for SDN.Software-Defined Networks (SDN) seems to be a promising network architecture solution which decouples control and data planes and standardizes their communication via an Open Flow protocol. The increase in demand of transaction in different devices require secure communication channel between transmitter and receiver. For the secure communication channel apply channel coding between sender and receiver end. The channel coding that is also known forward error coding , Block coding and convolutional coding schemes. For the decoding of encoded message use viterbi decoder. For the simulation of proposed viterbi decoder use Xilinx 14.1 and for synthesis of proposed design use I-sim. The proposed design successful synthesis and simulated without any error and proposed design shows low area in terms of number of slices , number of flip flops, and number of LUTs, also calculate the delay that is very low.**

***Keywords*— Branch Metric Unit, Path Metric Unit, Survivor Memory unit, look up tables (LUTs), number of slices, number of Gates and delay, etc.**

I. INTRODUCTION

The redundancy allows the receiver to detect a limited number of errors that may occur anywhere in the message, and often to correct these errors without re-transmission. FEC gives the receiver the ability to correct errors without needing a reverse channel to request re-transmission of data, but at the cost of a fixed, higher forward channel bandwidth. FEC is therefore applied in situations where re-transmissions are costly or impossible, such as one-way communication links and when transmitting to multiple receivers in multicast. For example, in the case of a satellite orbiting around Uranus, a re-transmission because of decoding errors can create a delay of 5 hours. FEC information is usually added to mass storage (magnetic, optical and solid state/flash based) devices to enable recovery of corrupted data, is widely used in modems, is used on systems where the primary memory is ECC memory and in broadcast situations, where the receiver does not have capabilities to request retransmission or doing so would induce significant latency.

FEC processing in a receiver may be applied to a digital bit stream or in the demodulation of a digitally modulated carrier. For the latter, FEC is an integral part of the initial analog-to-digital conversion in the receiver. The Viterbi decoder implements a soft-decision algorithm to demodulate digital data from an analog signal corrupted by noise. Many FEC coders can also generate a bit-error rate (BER) signal which can be used as feedback to fine-tune the analog receiving electronics.

1.2 Forward Error Correction (Channel Coding)

Forward Error Correction (channel coding) is a powerful technique for increasing the transmission system margin. For example, with the standardized Reed-Solomon code used in submarine systems [8], a BER of lower than 10−13 can be obtained for a BER before correction of only 10−4, thus providing a 5.8-dB system margin.

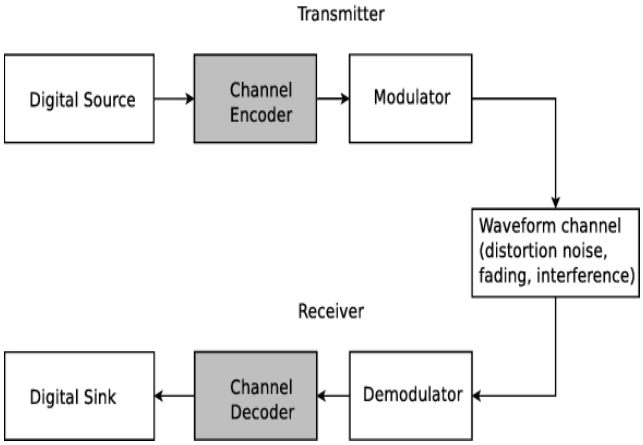

Fig 1.1 basic model of a digital transmission system using FEC techniques.

1.3 FEC Convolution Encoder

In telecommunication, a convolution code is a type of error-correcting code that generates parity symbols via the sliding application of a Boolean polynomial function to a data stream. The sliding application represents the 'convolution' of the encoder over the data, which gives rise to the term 'convolution coding'. The sliding nature of the convolution codes facilitates trellis decoding using a time-invariant trellis. Time invariant trellis decoding allows convolution codes to be maximum-likelihood soft-decision decoded with reasonable complexity.

The ability to perform economical maximum likelihood soft decision decoding is one of the major benefits of convolution codes. This is in contrast to classic block codes, which are generally represented by a time-variant trellis and therefore are typically hard-decision decoded. Convolution codes are often characterized by the base code rate and the depth (or memory) of the encoder [n , k , K] {\displaystyle [n,k,K]}. The base code rate is typically given as n / k {\displaystyle n/k}, where n {\displaystyle n} is the input data rate and k {\displaystyle k} is the output symbol rate. The depth is often called the "constraint length"K {\displaystyle K}, where the output is a function of the current input as well as the previous K − 1 {\displaystyle K-1} inputs. The depth may also be given as the number of memory elements v {\displaystyle v} in the polynomial or the maximum possible number of states of the encoder (typically : 2 v {\displaystyle 2^{v}}).

Convolution codes are often described as continuous. However, it may also be said that convolution codes have arbitrary block length, rather than being continuous, since most real-world convolution encoding is performed on blocks of data. Convolution ally encoded block codes typically employ termination. The arbitrary block length of convolution codes can also be contrasted to classic block codes, which generally have fixed block lengths that are determined by algebraic properties.

II. LITERATURE SURVEY

Chandel, S. et.al. (2019) - In this presented work authors presented Viterbi algorithm. It's the remarkably algorithm to decode the convolution code. The main drawback is its high computational complexity and power hungry implementation for large coding rate since its constraint length should be high. In this work presented, Viterbi decoder with modified Branch metic calculation is designed in order to decrease the hardware usage and to simplify the proceedings **[01].**

Rawoof, M. A.,et.al. (2019) - In this presented work authors presented the Viterbi algorithm. It's interesting as well as challenging for the researchers in the field of communications. It also has a broader range of applications in the digital communications field in this modern era of communications. This work helps in making use of efficient coding and decoding techniques with help of Viterbi algorithm. Also Viterbi algorithm can be easily understood and can be implemented easily. This work presents the implementation of the Viterbi algorithm using Verilog coding. Unlike other algorithms the proposed Viterbi algorithm has many advantages such as power consumption and the major advantage is error correction using Verilog **[02].**

Taotao, Z.,et.al. (2019, June) - In this presented work authors presented convolution code with large constraint length. It's plays an irreplaceable role in deep space communication and ultra-low frequency communication. Therefore, it is very important to find and test convolution code with large constraint length. Convolutional codes are widely used in deep space communication systems because of their high coding gain and simple and reliable encoders. The performance and implementation difficulty of convolutional codes mainly depend on the constraint length of the decimal codes and the coding efficiency. Enlarging and improving the coding gain of convolutional codes will greatly increase the complexity of decoders. The method of mathematical derivation and verification by mathematical tools is not suitable for convolutional codes with large large constraint length.. In order to improve the efficiency of inspection, the method of parallel multi-core computing needs to be introduced into the evil code test. Tests show that the FPGA-based parallel inspection method can improve the test efficiency by geometric multiples **[03].**

Giri, S. D., et.al. (2019, February) - In this presented work authors presented design of the structure of Convolutional code to reduce the influence from multi path and channel noise. Such a system can do flexibility relevancy ever-changing information rates, increasing vary, and increasing diversity, whereas giving economical resource utilization. This design is capable of transmitting data, in air errors and noise are tried to be minimized by using channel coding technique. The data speed can be increased by using different combinations of encoding and modulation techniques **[04].**

Harsh, G. B., et.al. (2019) -In this presented work authors presented by analysis the affiliation amid the aphorism computations, a different alignment was projected, that is called MSR. By applying the projected alignment to the antecedent ACS architectures, Associate in nursing area-efficient architecture for algebraic computations was achieved. The projected architectures attain at the a lot of eighteen.1% abridgement in superior in befitting with the accomplishing after-effects that appreciably reduces the superior of the abounding MAP amount of the turbo decoder. What is more, the projected alignment may be acclimated for college abject styles to cut aback quality **[05].**

III. PROPOSED METHOD

3.1 Viterbi Algorithm

The Viterbi algorithm has been known as a maximum likelihood decoding algorithm for convolutional codes. Let us consider a simple example for illustrating the principle of Viterbi algorithm. Assume that a car that has 3 states forward, stop and reverse with the condition that a transition from forward to reverse is not allowed. In other words, it implies that the car first enter the stop state and then enter the reverse state. Hence, when we receive the information through the processes of forward, reverse and stop, we can safely interpret it as – forward, stop and reverse as this is a "maximum likelihood sequence". The Viterbi algorithm uses the trellis diagram to compute the path metric value (accumulated distance) from the received sequence to the possible transmitted sequences. The total number of such trellis paths increases exponentially with the number of stages in the trellis. It causes potential complexity and memory problems. The Viterbi decoding algorithm has been classified into hard decision decoding and soft decision decoding. If the received signal is converted into two levels, either zero or one, it is called hard decision. If the input signal is quantized and processed for more than two levels, it is called soft decision. The soft decision decoding is expensive and require large amount of memory than hard decision decoding. Hence, this work focuses on the hard decision decoding. Figure 5.1 shows the trellis diagram for hard decision Viterbi decoding. When a sequence of data is received from the channel, it is desirable to estimate the original sequence that has been sent. The process of identifying such a sequence can be done using a diagram called 'trellis',

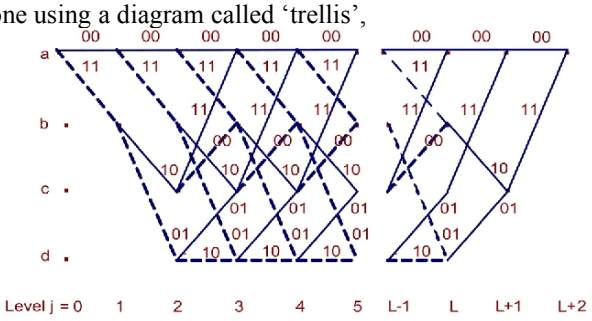

Fig 3.1 Viterbi decoding trellis diagram

The detection of the original stream can be described as finding the most probable path through the trellis. In the trellis diagram each node specifies an individual state at a given time and indicates a possible pattern of recently received data bits. The transition to a new state at the next timing cycle is indicated by each branch.

3.2 Viterbi Decoder

Viterbi algorithm is used in the Viterbi decoder for decoding a bit stream that has been encoded using FEC based on a Convolutional code. Figure 5 shows the block diagram of Viterbi decoder. It consists of the following functional units, namely, Branch Metric Unit, Path Metric Unit, Survivor Memory unit.

Branch Metric Unit

A branch metric unit's function is to calculate branch metrics, which are normed distances between every possible symbol in the code alphabet, and the received symbol.

There are hard decision and soft decision Viterbi decoders.

A hard decision Viterbi decoder receives a simple bitstream on its input, and a Hamming distance is used as a metric. A soft decision Viterbi decoder receives a bitstream containing information about the reliability of each received symbol. For instance, in a 3-bit encoding, this reliability information can be encoded as follows:

Table 3.2 Shows a 3-bit encoding

value	meaning	
000	strongest	**0**
001	relatively strong	**0**
010	relatively weak	**0**
011	weakest	**0**
100	weakest	**1**
101	relatively weak	**1**
110	relatively strong	**1**
111	strongest	**1**

Of course, it is not the only way to encode reliability data. The squared Euclidean distance is used as a metric for soft decision decoders.The comparison between received code symbol and expected code symbol is done by branch metric unit. It also counts the number of differing bits. It is the smallest unit in the Viterbi decoder. The measured value of the BMU can be the Hamming distance in case of the hard input decoding or the Euclidean distance in case of the soft input decoding.

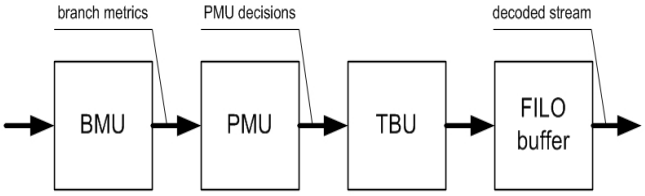

Fig 3.2 a. A common way to implement a hardware viterbi decode Branch metric unit (BMU)

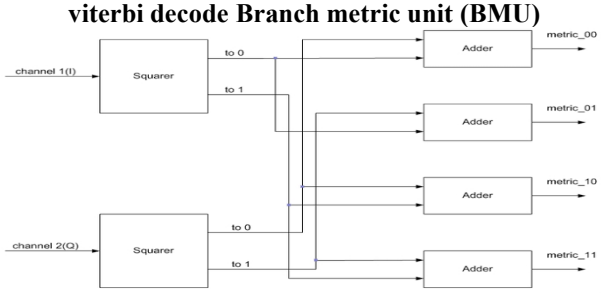

Fig 3.2 b. A sample implementation of a branch metric unit

B. Path Metric Unit

The partial path metric of each branch is computed by the use of two adders. The partial path metric is compared by the comparator and an appropriate branch is selected by the selector. The selector selects the smaller value and stored. A path metric unit summarizes branch metrics to get metrics for 2 K − 1 {\displaystyle 2^{K-1}} paths, where K is the constraint length of the code, one of which can eventually be chosen as optimal. Every clock it makes 2 K − 1 {\displaystyle 2^{K-1}} decisions, throwing off wittingly nonoptimal paths. The results of these decisions are written to the memory of a traceback unit.

The core elements of a PMU are ACS (Add-Compare-Select) units. The way in which they are connected between themselves is defined by a specific code's trellis diagram.

Since branch metrics are always ≥ 0 {\displaystyle \geq 0} , there must be an additional circuit (not shown on the image) preventing metric counters from overflow. An alternate method that eliminates the need to monitor the path metric growth is to allow the path metrics to "roll over"; to use this method it is necessary to make sure the path metric accumulators contain enough bits to prevent the "best" and "worst" values from coming within 2(n-1) of each other. The compare circuit is essentially unchanged.

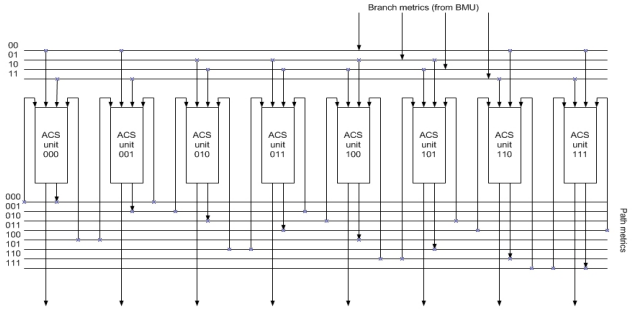

Fig 3.3 A sample implementation of a path metric unit for a specific K=4 decoder

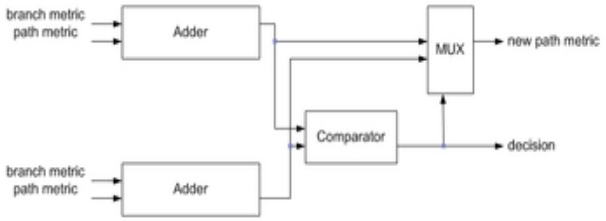

Fig 3.4 A sample implementation of an ACS unit

It is possible to monitor the noise level on the incoming bit stream by monitoring the rate of growth of the "best" path metric. A simpler way to do this is to monitor a single location or "state" and watch it pass "upward" through say four discrete levels within the range of the accumulator.

C. **Trace Metric Unit / Trace Back Unit**

Back-trace unit restores an (almost) maximum-likelihood path from the decisions made by PMU. Since it does it in inverse direction, a viterbi decoder comprises a FILO (first-in-last-out) buffer to reconstruct a correct order. Note that the implementation shown on the image requires double frequency. There are some tricks that eliminate this requirement.

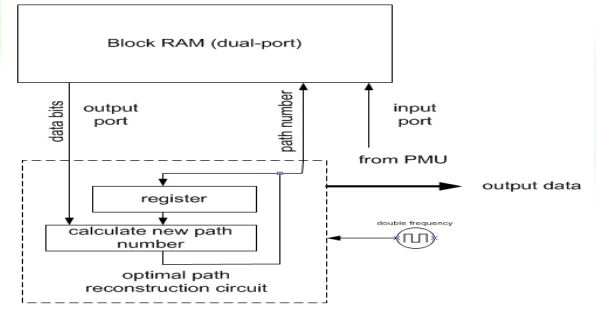

Fig 3.5 A sample implementation of a traceback unit

The general approach to traceback is to accumulate path metrics for up to five times the constraint length (5 (K - 1)), find the node with the largest accumulated cost, and begin traceback from this node.

The commonly used rule of thumb of a truncation depth of five times the memory (constraint length K-1) of a convolutional code is accurate only for rate 1/2 codes. For an arbitrary rate, an accurate rule of thumb is 2.5(K - 1)/(1−r) where r is the code rate.

However, computing the node which has accumulated the largest cost (either the largest or smallest integral path metric) involves finding the maxima or minima of several (usually 2K-1) numbers, which may be time consuming when implemented on embedded hardware systems.

D. State Metric Unit

The simplest method to increase throughput is to decompose the input stream into blocks of length which can be processed in parallel using conventional Viterbi decoders. Such an architecture yields a -fold increase in throughput for a -fold increase in complexity. However, independent block processing requires knowledge of the initial state metrics which are unknown until the previous block is processed. Hence, without additional information,

this approach is of no practical use because is limited to one. Two practical approaches to block-based decoding are state initialization and interleaving. State initialization achieves block independence by forcing the encoder to a known state at the start of each block. This technique reduces the information rate and adds complexity to the decoder which must obtain block synchronization from the received data. Interleaving achieves block independence by interleaving independently encoded sequences. By definition, the blocks are independent and hence can be decoded independently; however, the encoded sequences are interleaved prior to the addition of noise.

IV. SIMULATION RESULT OF PROPOSED WORK

4.1 Introduction

This chapter describes the simulation and analysis of the results of the proposed method. The results describe by graphs and tables and also compare the results of the projected works with the previous works.

The basic configuration of our system is core i5 4GB RAM. The performances are quantitatively measured by the proposed decoder architecture considering the use of resources and the speed as defined in given equations. Also compare on the basis of the comparison of the proposed decoder architecture by considering the throughput, latency and surface area means the total number of gates used in the FPGA structure.

The Viterbi decoder takes the distance measures and calculates the most likely transmitted signal. It does this by keeping a running history of the previously received signals in a path memory.

The path-memory length of this decoder is 12. By keeping a history of possible sequences and using the knowledge that the signals were generated by a state machine, it is possible to select the most likely sequences.

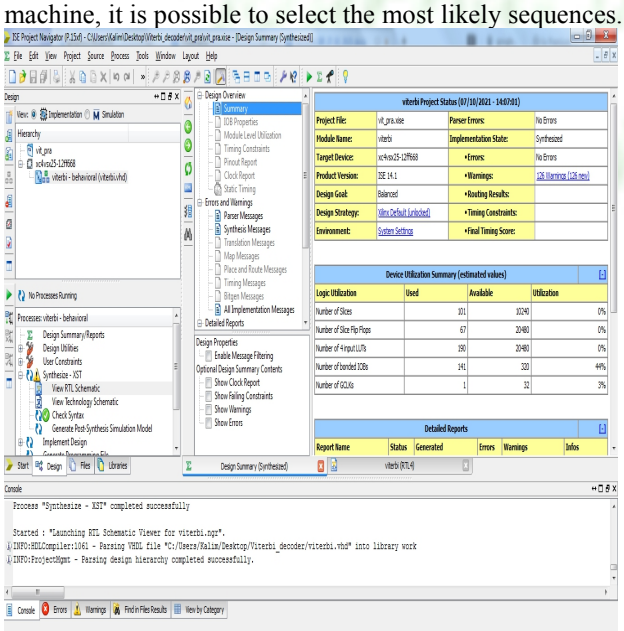

Fig. 4.5 Viterbi Summery Report

RTL View

In the above section discuss the design summery of proposed viterbi decoder now discuss about the RTL schematics of viterbi decoder, in the RTL schematics shows input and output of proposed design IC. In the proposed design as a input used data. clock and reset. The input data is encoded data that is decoded by viterbi decoder and shows the decoded data in data_out.

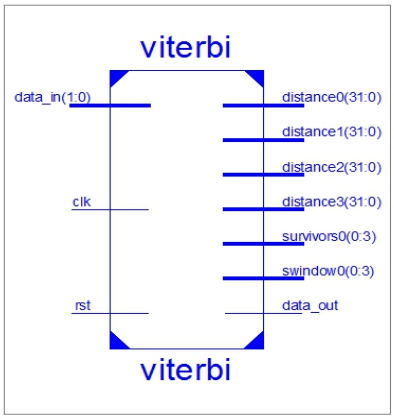

Fig. 4.6 RTL Schematics

In the above chapter discuss about the logic level implementation of viterbi decoder also discuss the number of used components of design, and discuss the logic delay. In the next chapter discuss about the Simulation of viterbi decoder.

I-SIM Simulator Result Wave Form

The system input or message, Data_in[1:0] , is driven by a counter that repeats the sequence 0, 3,2,0,1,0,1,3,1,0 ... by random number at each positive clock edge (with a delay of one time unit), starting with X equal to 4 at t= 0. The active-high reset signal, Res, is asserted at t = 40. The encoder output, uncoded_signal[2:0], changes at t = 60, which is one time unit (the positive-edge-triggered D flip-flop model contains a one-time-unit delay) after the first positive clock edge (at t = 99) following the dissertation of the reset at t = 1000ns. The encoder output sequence beginning at t= 62 is and then the sequence 0, 3,2,0,1,0,1,3,1,0 ... . This encoder output sequence is then imagined to be transmitted and received. The Viterbi decoder model presented in this work is written for both simulation and synthesis. The Viterbi decoder makes extensive use of vector D flip-flops (registers). Early versions of VHDL did not support vector instantiations of modules. In addition the inputs of UDPs may not be vectors and there are no primitive D flip-flops in VHDL. This makes instantiation of a register difficult other than by writing a separate module instance for each flip-flop.

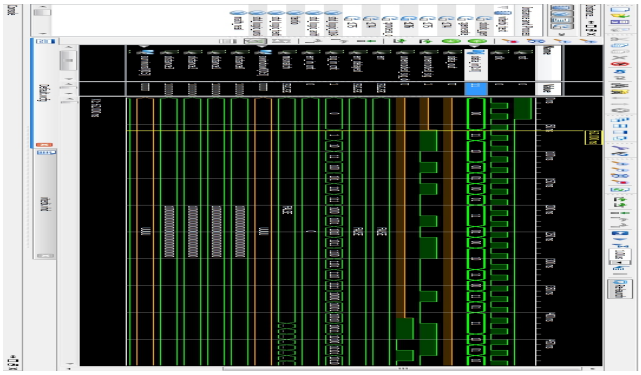

Fig. 4.7 Shows the I-Sim Simulator output

Instance and Process Name	Design Unit	Block Type
viterbi_test	viterbi_test(b...	VHDL Entity
std_logic_1164	std_logic_1164	VHDL Package
std_logic_arith	std_logic_arith	VHDL Package
std_logic_signed	std_logic_sig...	VHDL Package
textio	textio	VHDL Package
std_logic_textio	std_logic_tex...	VHDL Package
std_logic_unsigned	std_logic_un...	VHDL Package
math_real	math_real	VHDL Package

Fig. 4.8 Shows the Different Lib. used in decoder

4.1 wave form random seed 1, random seed 2, random seed 3 and random seed 4.

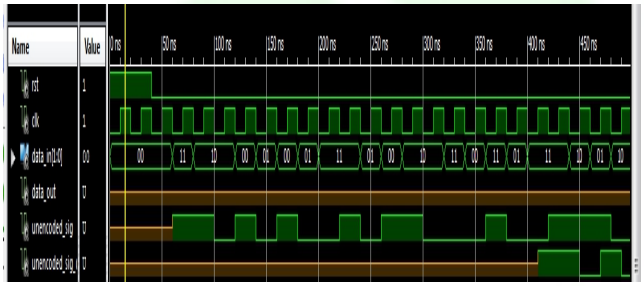

Fig. 4.9 Shows the Input data in Binary form (Wave Form)

the above figure 4.9 data_in[1:0] shows the input signal and data encoder wave form to the proposed encoder.

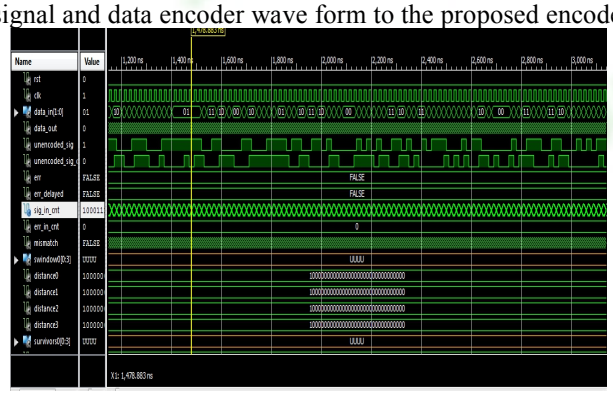

Fig. 4.10 shows the resultant decoder output

In the above figure 5.6 shows the decoder output resultant.

4.6 Result Comparison

the above section discuss the simulation outcomes on i-sim simulator now discuss the result comparison of proposed decoder compare with previous decoder. That is shown in below table 4.1.

Table 4.1 Comparison Of Hardare Ulitization

S.No.	Existing Methods	Device	Number of LUT Slice used
1	[2]	Vertix 4	172
2	[.3]	Vertix 4	131
3	Proposed	Vertix -4	101

V. CONCLUSION

In this research work first discussed on different viterbi decoder that is presented by different researchers in the last decade, describe in the literature survey. Also discuss the viterbi decoder and parts in technical background. The proposed viterbi doeoder is design by three different parts Branch Metric Unit, Path Metric Unit, Survivor Memory unit shown in the decoder.

For the simulation of proposed viterbi decoder use Xilinx 14.1 and for synthesis of proposed design use I-sim. The proposed design successful synthesis and simulated without any error and proposed design shows low area in terms of number of slices , number of flip flops, and number of LUTs, also calculate the delay that is very low. These three parameters i.e. power, area and speed are always traded off. However, area and speed are usually conflicting constraints, so that improving speed results mostly in larger areas. In the result shows the comparison of proposed work with base paper on different result parameters such as look up tables (LUTs), number of slices, number of Gates and delay and shows better result as compare to previous work.

REFERENCES

[1]. Chandel, Suman, and Manju Mathur. "Viterbi Decoder Plain Sailing Design for TCM Decoders." (2019).

[2]. Rawoof, Md Abdul, et al. "Verilog based efficient convolution encoder and viterbi decoder." International Journal of Reconfigurable and Embedded Systems 8.1 (2019): 75.

[3]. Taotao, Zhang, et al. "FPGA-Based Large Constraint Length Convolution Code Encoder Verification." Journal of Physics: Conference Series. Vol. 1237. No. 4. IOP Publishing, 2019.

[4]. Giri, S. D., P. P. Patil, and P. B. Murmude. "Performance Analysis of Convolutional encoder based on FPGA." 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT). IEEE, 2019.

[5]. Harsh, G. Bhuvanendra, and G. Lakshma Reddy. "A VLSI Design of LTE Turbo Encoder-Decoder with Radix 4 ACS Architecture." (2019).

[6]. Sharma, Vibhuti, and Sunil Sharma. "Analyzing the Bit Error Rate & Hardware Implementation of Convolution Encoder & Viterbi Decoder."(2019)

[7]. Sujatha, E., C. Subhas, and Giri Prasad. "Performance improvement of Turbo Decoder using VLSI Optimization Techniques." 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN). IEEE, 2019.
[8]. Gao, Zhen, et al. "Design and Implementation of Configuration Memory SEU-Tolerant Viterbi Decoders in SRAM-Based FPGAs." IEEE Transactions on Nanotechnology 18 (2019): 691-699.
[9]. Amarnath, V., et al. "Simulation Improvement of Rising Field Programmable Gate Array Rectify Methodologies by Victimisation Playacting SEC-DAED-TAED-TETRA AED Code." Journal of Computational and Theoretical Nanoscience 16.5-6 (2019): 2362-2367.
[10]. Сайлауқызы, Ж., and М. Коккоз. "Research of Noise Immunity of Viterbi Decoder in the Case of Different Depths of Decoding in MATLAB Environment and Projection on FPGA." (2019).
[11]. Ahmadinejad, Hosein, Abolfazl Falahati, and Ebrahim Shafiee. "Design and Implementation of a Visible Light Communication and Coding System Based on IEEE802. 15.7 Standard." 2019 2nd West Asian Colloquium on Optical Wireless Communications (WACOWC). IEEE, 2019.
[12]. Zbaid, Riham Ali, and Kasim K. Abdalla. "Design and Implementation of Convolutional Encoder and Viterbi Decoder Using FPGA." Journal of University of Babylon 26.3 (2018): 22-29.
[13]. Kumari, Dasari Ratna, G. Srinivasa Rao, and K. Anka Siva Prasad. "Low Cost VLSI Architecture for Proposed Adiabatic Offset Encoder and Decoder." (2018).
[14]. Thakur, Akash, and Manju K. Chattopadhyay. "Design and Implementation of Viterbi Decoder Using VHDL." IOP Conference Series: Materials Science and Engineering. Vol. 331. No. 1. IOP Publishing, 2018.
[15]. Prasad, N., Indrajit Chakrabarti, and Santanu Chattopadhyay. "An energy-efficient network-on-chip-based reconfigurable Viterbi decoder architecture." IEEE Transactions on Circuits and Systems I: Regular Papers 65.10 (2018): 3543-3554.
[16]. Adiono, Trio, Ahmad Zaky Ramdani, and Rachmad Vidya Wicaksana Putra. "Reversed-Trellis Tail-Biting Convolutional Code (RT-TBCC) Decoder Architecture Design for LTE." International Journal of Electrical & Computer Engineering (2088-8708) 8.1 (2018).
[17]. Tobola, John D., and James E. Stine. "Low-Area Memoryless optimized Soft-Decision Viterbi Decoder with Dedicated Paralell Squaring Architecture." 2018 52nd Asilomar Conference on Signals, Systems, and Computers. IEEE, 2018.
[18]. Xiaobo, Jiang, Zhang Fang, and Zeng Zhen. "High-performance Decoder for Convolutional Code with Deep Neural Network." arXiv preprint arXiv:1812.11455 (2018).
[19]. Yueksel, Hazar, et al. "Design Techniques for High-Speed Multi-Level Viterbi Detectors and Trellis-Coded-Modulation Decoders." IEEE Transactions on Circuits and Systems I: Regular Papers 65.10 (2018): 3529-3542.
[20]. Adiono, Trio, et al. "Design of an OFDM system for VLC with a Viterbi decoder." IEIE Transactions on Smart Porcessing and Computing (SPC) 6.6 (2017): 455-465.
[21]. Laddha, Deepali R., Archana O. Vyas, and M. E. Student. "FPGA Implementation of Viterbi Decoder for Long Survivor Path." International Journal of Engineering Science 11822 (2017).
[22]. Wankhede, Shweta Anand, and Nilesh Bodne. "Review Paper On Implementation Of Low Power Hard Decision Viterbi Decoder In VLSI." (2017).
[23]. Shende, Ms Sneha T., and Asst Prof SK Tadse. "Designing of Asynchronous Viterbi Decoder for Low Power Consumption using Handshaking Protocol: A Technical Review." (2017).

