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

Volume-14, Issue-04, April 2025
JOURNAL OF COMPUTING TECHNOLOGIES (JCT)
International Journal
Page Number: 01-06

A Review of Literature on Convolution Encoder and Viterbi Decoder and Its Various Aspect

Polarati Gayatri [1], Aditya Mishra [2]
[1,2] VLSI M.Tech Scholar, Assistant Professor
[1,2] Department of Electronics and Communication Engineering
[1,2] Vidhyapeeth Institute of Science & Technology Bhopal, RGPV, (M.P.), INDIA
Gmail: gayatripolarati123@gmail.com [1] adivgi023@gmail.com [2]

***Abstract*—Error detection and correction are critical components in modern digital communication systems to ensure data integrity across noisy channels. Convolutional encoding and Viterbi decoding represent a widely adopted pair of techniques used to enhance the reliability of data transmission. This literature review presents a comprehensive analysis of convolutional encoders and Viterbi decoders, exploring their historical development, working principles, algorithmic structure, and various implementations. Key parameters such as constraint length, code rate, and generator polynomials are discussed in relation to their impact on encoding efficiency and decoding complexity. The review also examines the Viterbi algorithm's evolution, including hard and soft decision decoding, reduced-state methods, and hardware-friendly architectures. Special emphasis is placed on performance trade-offs between decoding accuracy, latency, and hardware resource consumption.**

Keywords*— *Convolutional Encoder, Viterbi Decoder, Error Correction, Forward Error Correction (FEC), Digital Communication, Soft Decision Decoding, Hard Decision Decoding etc.

I. INTRODUCTION

The receiver may identify a limited number of faults anywhere in the message and typically rectify them without retransmission due to redundancy. At the expense of a fixed, greater forward channel capacity, FEC allows the receiver to repair mistakes without requesting retransmission. One-way communication lines and multicast broadcasts to numerous recipients need FEC since retransmissions are expensive or impractical. An Uranus-orbiting satellite's re-transmission due to decoding problems may take 5 hours. FEC information is added to mass storage (magnetic, optical, and solid state/flash-based) devices to recover corrupted data, is widely used in modems, on systems with ECC memory, and in broadcast situations where the receiver cannot request retransmission or would incur significant latency. A receiver may demodulate a digital carrier or process a digital bit stream using FEC. The receiver's first analog-to-digital conversion includes FEC. Viterbi decoders use soft-decision algorithms to demodulate noise-corrupted analog signals into digital data. Bit-error rate (BER) signals from several FEC coders may be utilized to fine-tune analog reception devices. Various forward error correcting codes are suited for various scenarios since the ECC design limits the maximum fractions of mistakes or missing bits that may be fixed. Stronger codes need more bandwidth to transmit redundancy, lowering the effective bit-rate and enhancing the received signal-to-noise ratio.

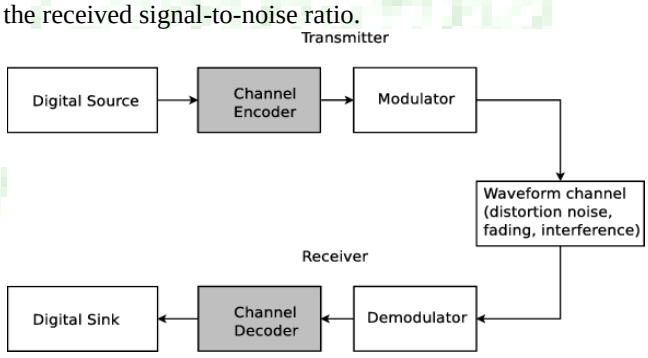

Fig 1. Basic model of a digital transmission system using FEC techniques

The channel encoder adds controlled redundancy to the binary information sequence to help the receiver detect and fix mistakes. In particular, the channel encoder converts k information symbols into a unique n-symbol code word. Code rate = k/n. The inverse of the code rate, n/k, measures encoding redundancy.

The reception decoder evaluates the encoder's rules to discover and repair transmission problems. This adjustment boosts coding. Reed Solomon codes, utilized in wireless,

satellite, magnetic, and optical recording, are block codes' most successful use. FEC was initially deployed in underwater transmission systems using RS(255,239). RS codes' non-binary architecture (RS encoder/decoder processes m-bit symbols at each clock top) make them ideal for large data rates. Simple RS encoder/decoder implementation is also possible. ITU-T G.975 and G709 use the RS(255,239) as a standard FEC for subsea and terrestrial optical fiber transmission systems. Convolution encoder 1.2 FEC Convolution codes are error-correcting codes that slide a Boolean polynomial function over a data stream to create parity signals. The sliding application depicts the encoder's 'convolution' across the data, thus 'convolution coding'. Time-invariant trellis decoding is possible because convolution codes slide. Convolution codes may be maximum-likelihood soft-decoded with moderate complexity using time-invariant trellis decoding. Convolution codes provide cost-effective maximum likelihood soft choice decoding. Traditional block codes, represented by a time-variant trellis, are hard-decision decoded. Base code rate and encoder depth define convolution codes. The base code rate is typically given as , where is the input data rate and is the output symbol rate. The depth is termed the "constraint length" since the output depends on the current and prior inputs. The depth may also be given as the number of memory elements in the polynomial or the maximum possible number of states of the encoder (typically :). Many call convolution codes 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. Termination is common in convolutional block codes. 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.

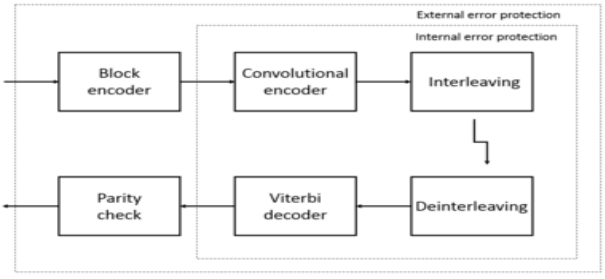

Fig 2 Stages of channel coding in GSM

Recursive and Non-Recursive Codes

The encoder on the picture above is a non-recursive encoder. Here's an example of a recursive one and as such it admits a feedback structure:

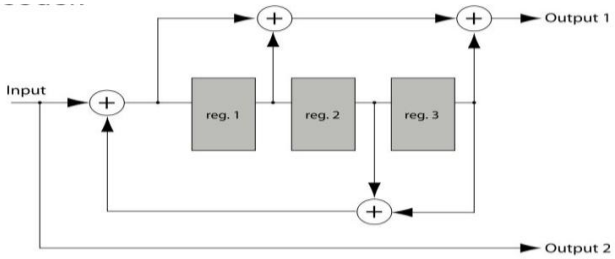

Fig. 3 Rate 1/2 8-state recursive systematic convolutional encoder

The example encoder is systematic because the input data is also used in the output symbols (Output 2). Codes with output symbols that do not include the input data are called non-systematic.

Recursive codes are typically systematic and, conversely, non-recursive codes are typically non-systematic. It isn't a strict requirement, but a common practice.

The example encoder in Img. 2. is an 8-state encoder because the 3 registers will create 8 possible encoder states (23). A corresponding decoder trellis will typically use 8 states as well.

Recursive systematic convolutional (RSC) codes have become more popular due to their use in Turbo Codes. Recursive systematic codes are also referred to as pseudo-systematic codes.

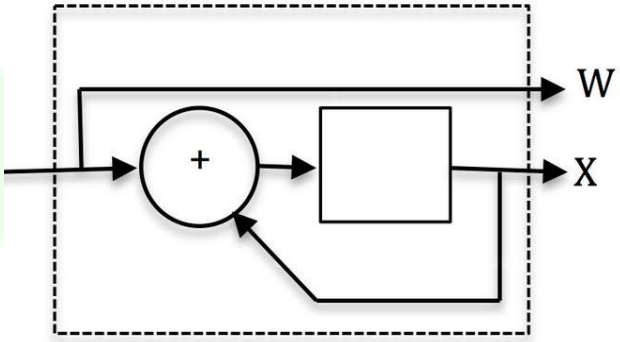

Fig. 4 Two-state recursive systematic convolutional (RSC) code

Trellis Diagram

A convolutional encoder is a finite state machine. An encoder with n binary cells will have 2n states.

Imagine that the encoder (shown on Img.1, above) has '1' in the left memory cell (m0), and '0' in the right one (m−1). (m1 is not really a memory cell because it represents a current value). We will designate such a state as "10". According to an input bit the encoder at the next turn can convert either to the "01" state or the "11" state. One can see that not all transitions are possible for (e.g., a decoder can't convert from "10" state to "00" or even stay in "10" state).

All possible transitions can be shown as below:

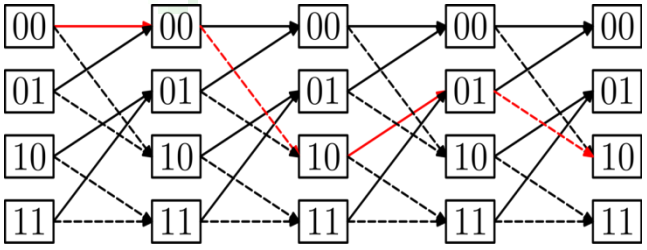

Fig. 5 A trellis diagram for the encoder on Img.1

II. LITERATURE REVIEW

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 presentedby 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].

Sharma, V., et.al. (2019) -In this presented work authors presented the Viterbi decoding algorithm is mature error correct system, which will give us a BER at 8.6E-007 at 5db on an AWGN channel with BPSK modulation. By puncturing, for rate 2/3, we will pay around a 2db cost. For rate 3/4, we will pay for a 3 db cost during the transmission. From the results, we find the Viterbi decoding algorithm is mature error correct system, which will give us a BER at 8.6E-007 at 5db on an AWGN channel with BPSK modulation. By puncturing, for rate 2/3, we will pay around a 2db cost. For rate 3/4, we will pay for a 3 db cost during the transmission. For the time issue, we do not implement a higher performance Viterbi decoder with such as pipelining or interleaving. So in the future, with Pipeline or interleave the ACS and the trace-back and output decode block, we can make it better [06].

Sujatha, E., et.al. (2019, March) -In this presented work authors presented Optimised MAP Turbo Decoder with Recursive QPP interleaver/Deinterleaver is simulated, synthesized and implemented using Xilinx Vivado 14.2 tool along with 28 nm CMOS Zynq Zed board FPGA device and the design results shows better performance to other conventional designs. Here, parallel computation of state metrics, branch metrics and intelligent memory scheduling for storage of intermediate metrics introduced at decoder level. Latency has been reduced by performing intelligent memory partitioning in turbo decoder. On-fly computation of address locations of interleaved data has been done in QPP interleaver to reduce the memory requirement. The designed optimized turbo decoder is low complex, simple and use single type and re-usable computing resource ACS units for various computations. By applying VLSI optimization techniques of parallel computation, intelligent memory partitioning 50% of the computation period is reduced such that the latency decreased to compute extrinsic information from LLRs [07].

Gao, Z., Zhu, et.al.. (2019) -In this presented work authors presented effects of soft errors on the configuration memory of a Viterbi decoder implemented on an SRAMFPGA. To that end, errors have been injected in the configuration memory, and the results show that the decoder is able to correct most of the errors when the BER is low or the SNR is high, but this immunity degrades significantly when the input signal deteriorates. Based on this first result, a new technique has been proposed to protect the Viterbi decoder against errors in the configuration memory. The scheme is based on an enhanced Duplication with Comparison (DWC) on which some internal signals are used to detect the copy in error so that upon a difference on the outputs, the ones from the correct copy can be used. The technique has been implemented and evaluated. The results show that the technique provides an efficient protection with a resource usage that is significantly lower than that of TMR.

Compared to the scheme, the proposed technique achieves a much better reliability with a similar overhead [08].

Amarnath, V.et.al. (2019) -In this presented work authors presented BER performance of SECCC is investigated with/without parallelism and with different frame sizes and also compared with the TC, where both schemes employ the same RSC code and code rate. In order to invoke a complete comparison, the VHDL design of MAP decoder for both schemes is synthesized using Xilinx ISE. The synthesis results show that both schemes produce equal throughput and exhibit equal resource utilization for the same number of iterations, frame sizes and parallelism. Based on the simulation results presented in Section IV-B it can be concluded that for BER of 10−4 , SECCC outperforms TC for frame sizes less than or equal to 2048 bits with parallelism of 16, 32 and 64 as well as for frame sizes greater than or equal to 6144 bits with parallelism of 256. However, Fig. 13 and Fig. 15 show that for achieving BER of 10−6 at parallelism of 32, 64, 128 and 256 with NII-method, TC still exhibits a small error floor. At higher parallelism the frame size is divided into smaller sized sub-frames and in case of SECCC, the single trellis is longer than each of the two trellises of TC, therefore SECCC performs better for smaller sized frames at higher parallelism [09].

Сайлауқызы, et.al. (2019) -In this presented work authors presented basis of theoretical and experimental research in the work, the problems are solved in the study of the Viterbi decoder of modern communication systems using convolutional codes. This paper discusses the architecture and details of the hardware implementation of a Viterbi decoder for a convolutional code with a basic coding speed 1/2, a constraint length K = 7, a predetermined by generator polynomials (133, 177) 8. In particular, a convolutional code (133, 177) 8 is used in the standard of DVB-T digital terrestrial television, as the inner code encoding circuit cascade[10].

III. CHELLENGES

The design and implementation of an enhanced Viterbi decoder using VHDL for Software-Defined Networks (SDN) is an ambitious and technical project. The Viterbi algorithm, primarily used for decoding convolutional codes in communication systems, can be optimized in various ways, especially in the context of SDN, where flexibility, performance, and scalability are key concerns.

1. Complexity of the Viterbi Algorithm
   - Challenge: The Viterbi algorithm involves dynamic programming and can be computationally intensive, especially as the constraint length of the convolutional code increases. Implementing this in VHDL requires careful management of resources, pipelining, and memory, as FPGA or ASIC implementations have limited resources.
   - Solution: Implement efficient memory management strategies, like using circular buffers or employing techniques like reduced-state Viterbi algorithms or partial-path storage.

2. FPGA/ASIC Resource Constraints
   - Challenge: FPGAs and ASICs have finite resources (logic gates, memory, etc.). A high-performance Viterbi decoder might require significant resources, especially when implementing features like parallel processing for decoding multiple streams simultaneously.
   - Solution: Optimize the Viterbi algorithm to minimize hardware usage. Consider resource-sharing techniques such as multiplexing hardware components, using smaller registers, and optimizing the decoding process.

3. Latency and Throughput Trade-Offs
   - Challenge: Viterbi decoding is often used in real-time systems, where low latency is critical. However, increasing throughput (parallelization or pipelining) might increase latency. Finding the right balance between these factors is tricky.
   - Solution: Implementing pipelining in the Viterbi algorithm or using multi-stage decoders might help achieve both low latency and high throughput. Fine-tuning pipeline depth and parallelism is key to optimizing performance.

4. Improved Performance in SDN Context
   - Challenge: In SDN, there's a need to integrate this Viterbi decoder with high-speed networking systems, where performance demands are critical. Network congestion and bandwidth issues might require even faster decoding algorithms or adjustments to make the decoder work efficiently within the SDN architecture.
   - Solution: Tailor the Viterbi decoder for SDN by focusing on packet-level decoding and ensuring it handles high-speed data transmission with minimal latency. Consider optimizing the decoder's performance in distributed SDN environments.

5. VHDL Code Optimization for Parallel Processing
   - Challenge: VHDL is not inherently high-level, and achieving a parallel implementation of the Viterbi algorithm can be complex. Parallel processing is crucial for improving the performance of the decoder but needs efficient VHDL coding practices.
   - Solution: Use parallel architecture for different parts of the Viterbi decoder (e.g., state metric calculation, path metric storage). Ensure VHDL constructs like process, signal, and variable are efficiently used for synchronization and data flow management.

IV. EXPECTED SOLUTION

Involves developing a high-performance, low-latency Viterbi decoder tailored for integration into Software-Defined Networking (SDN) environments. This solution requires the design of a convolutional encoder and a corresponding Viterbi decoder using VHDL, targeting

FPGA implementation to ensure real-time decoding capabilities. The Viterbi decoder should incorporate architectural optimizations such as pipelining, parallel processing, and memory-efficient traceback mechanisms to enhance decoding speed and throughput while reducing resource usage. Techniques like soft-decision decoding and reduced-state trellis optimization may also be applied to improve error correction accuracy without significantly increasing complexity. The decoder is expected to interface dynamically with an SDN controller, allowing real-time reconfiguration based on network conditions (e.g., packet error rates, bandwidth fluctuations). Performance metrics such as Bit Error Rate (BER), latency, and hardware resource utilization (LUTs, flip-flops, memory blocks) will be evaluated through simulation and synthesis using tools like Xilinx Vivado or Intel Quartus. Ultimately, this project aims to produce a VHDL-based Viterbi decoder that meets the high-speed, flexible, and programmable requirements of modern SDN architectures, enabling more reliable and adaptive data communication.

V. Conclusion

As described in the literature review, this study first examined viterbi decoders reported by researchers in the recent decade. Include technical background on the viterbi decoder and components. The viterbi decoder has three parts: Branch Metric Unit, Path Metric Unit, and Survivor Memory Unit. Use Xilinx 14.1 to simulate viterbi decoder and I-sim to synthesize design. The suggested design was synthesized and simulated without errors. It has a low area in terms of slices, flip flops, and LUTs, and a low latency. Power, area, and speed constantly trade off. Area and speed are frequently opposing restrictions, hence speed improvement mainly occurs in bigger regions. Comparing the proposed work to the base study on look up tables (LUTs), number of slices, number of gates, and latency provides improved results. Implementing convolution encoder and viterbi decoder with coding rate 2/3 in compact VHDL was the goal. VHDL implementation is module-based and simpler than other Verlog languages. Second, viterbi decoders are receiver-end designs, saving memory. Lookup approach reduces time and complexity while tracing route back to front end. Look up tables (LTUs), slices, flip flops, and global clock are improved with the suggested decoder. GCLCK.

REFERENCES

[1.] Namratha, & Bakhar, M. (2023). Power and area optimized adaptive Viterbi decoder for high speed communication applications. International Journal of Information Technology, 15(1), 45-52.

[2.] Devi, T. Kalavathi, E. B. Priyanka, P. Sakthivel, and A. Stephen Sagayaraj. "Low complexity modified viterbi decoder with convolution codes for power efficient wireless communication." Wireless Personal Communications 122, no. 1 (2022): 685-700.

[3.] Rowshan, M., & Viterbo, E. (2021). List Viterbi decoding of PAC codes. IEEE Transactions on Vehicular Technology, 70(3), 2428-2435.

[4.] Sokjabok, Siwakon, Chanon Warisarn, Santi Koonkarnkhai, and Jaejin Lee. "Modified 2D Viterbi algorithm using 2D modulation encoding constraints in BPMR systems." In 2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), pp. 457-460. IEEE, 2020..

[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 M. 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).
[24.] Kadu, Sanket, et al. "Design and Implementation of Viterbi Encoder and Decoder on FPGA." International Journal 3 (2017): 30-33.
[25.] Wankhede, Shweta Anand, and Nilesh Bodne. "A Configurable and Low Power Hard-Decision Viterbi Decoder in VLSI Architecture." (2017).
[26.] Elwazan, Aly AE, A. A. A. Zekry, and H. L. A. Zayed. "Matlab Code for LTE Convolutional Code and Viterbi Decoder." International Journal of Engineering Research & Technology (IJERT) 6.3 (2017): 578-581.
[27.] Nanthini, S., et al. "An Efficient Low Power Convolutional Coding with Viterbi Decoding Using FSM." Asian Journal of Applied Science and Technology (AJAST) Volume 1 (2017).
[28.] Rocha, Leandro MG, et al. "Physical implementation of an ASIC-oriented SRAM-based viterbi decoder." 2017 24th IEEE International Conference on Electronics, Circuits and Systems (ICECS). IEEE, 2017.
[29.] Jose, Jeena, and David Solomon George. "Design and implementation of a XOR-Free Lightweight Crypto-Coder System." (2017).
[30.] Buzdar, Abdul Rehman, et al. "Area and energy efficient Viterbi accelerator for embedded processor datapaths." Int. J. Adv. Comput. Sci. Appl. 8.3 (2017): 402-407.

