*IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)*
*Volume 14, Issue 5 (Sep. - Oct. 2024), pp.15-20*
*e-ISSN: 2319-4200, p-ISSN: 2319-4197*
*www.iosrjournals.org*

# Ternary Based System for Convolutional Encoder

Mayuri Chetan Rathi
*Research Scholar*
*Department of Electronics Engineering*
*University of Amravati*

Dr C. N. Deshmukh
*Professor*
*Department of Electronics Engineering*
*University of Amravati*

***Abstract -*** *Convolution coding has been used in communication systems including deep space communication and wireless communication. With demand for higher data rates beyond 10Gbps as per the recommendations of 60GHz communication standards processing speed of hardware and software modules need to further enhance to meet the requirements. VLSI circuits are designed with transistors sizing in nanometer range so as to operate at higher frequencies. With advances in semiconductor technology, transmitter and receiver circuits that meet the next generation wireless standards can be implemented on hardware that can operate at GHz frequency meeting the data rate greater than 10Gbps. Ternary computing uses trits to enable denser numerical encodings, support a fuzzy state, and give the cryptographer the opportunity to introduce new ciphering methods based on hardware primitives that can provide additional security. The focus is to search for new and better ternary convolutional codes with large free distance so more errors can be detected and corrected. The FEC code is used to secure data and information sent over the channel for storage as well as recover even in the presence of noise (errors). Interestingly, ternary architectures have been considered in computing and storage systems due to their higher radix economy and the three usable states for certain electromagnetic materials which have the potential to improve the performance of communication systems.*

***Index Terms –*** *Ternary computing, Ternary convolutional encoder , Higher frequency .*

## I. INTRODUCTION

Ternary logic which employs ternary arithmetic operations, ternary logic circuits and ternary memory is a promising alternative to binary logic as it can provide energy efficiency due to the reduced circuit overhead as compared to binary logic and requires fewer chip interconnections and less area . Ternary arithmetic operations and logic circuits are faster and require less energy than binary arithmetic operations and logic circuits. With integrated circuits, the number of interconnections required often determines the complexity. Ternary memory cells have a higher storage density than binary memory cells. Ternary phase shift keying (TPSK) modulation provides better error rate performance compared to binary phase shift keying (BPSK). This ever-growing demand for mobile data creates significant demands on the scarce wireless spectrum. Thus, improving the spectral and energy efficiency of wireless systems is an important challenge for the research community and industry. To address this issue, communication systems must efficiently utilize the available channel bandwidth by dynamically adjusting the transmission rate according to the channel conditions. Thus, to better use the available bandwidth, one should go beyond binary coding and modulation. Although ternary communication and computing systems have not yet reached commercial viability, ternary convolutional coding with ternary modulation has been shown to outperform existing conventional communication schemes with comparable computational complexity. Although ternary systems outperform binary systems in term of processing speed, wiring complexity and energy consumption, wireless communication systems today operate with binary data.. A Ternary convolutional code (TCC) can be represented by the parameters (n; k; m) where k and n are the numbers of input and output bits, respectively, and m is the encoder memory size. The code rate is k/n. Three code rates, ½, 1/3 and ⅔, are employed in the 802.11 standard.

. A ternary system has several important advantages over a binary one. These advantages can be summarized as: reductions in the interconnections required to implement logic functions, thereby reducing chip area; more information can be transmitted over a given set of lines; less memory requirement for a given data length. Besides this, serial and some serial-parallel operations can be carried out at higher speed. Its advantages have been confirmed in the application like memories, communications, and digital signal processing. Binary arithmetic is predominant in digital systems. Computation using binary arithmetic is simple, as binary devices have two states. Most computers today are based on binary logic. Ternary arithmetic operations and logic circuits are faster and require less energy than binary arithmetic operations and logic circuits. The main

contribution of this paper is to propose new ternary based convolutional codes, so we employ ternary convolutional coding to protect the ternary digit directly.

## II. DESIGN OF TERNARY CONVOLUTIONAL ENCODER

A convolutional encoder convolutes the incoming sequence of ternary traits into convolutional codes by adding more trits. The convolutional encoder takes k massage bit at a time from incoming sequence of trits stream and computes n-bit output sequence (n >k). The input to the convolutional encoder is nothing but randomizer output. A convolutional encoder consists of memory elements called shift registers and modulo 3 adders. For each k bit input, n output bits are calculated according to the generator polynomial. The computation is performed by module 3 addition operations on the current input k-bit symbol and the contents of the shift registers. The choice of connection between the shift registers and adders describes the characteristics of code. At the initial all shift registers are set by zeros. After performing one operation the contents of shift registers are shifted by one bit right. It is easy to construct a convolutional encoder. We first draw m boxes representing the m memory registers. Then we draw n modulo-3 adders representing the n output bits. Finally, we connect the memory registers to the adders using the bits specifying the generator polynomials. Shown in Fig. 1 is a (2, 1, 3) convolutional encoder. This encoder is going to be used to encode the 2-bit input sequence [00,01,10] with the two generator polynomials specified by the bits [1 1 2] and [0 2 1]. u represents the input bit, and G1 and G2 represent the output bits 1 and 2 respectively. m0 and m1 and m2 represent the initial state of the memory registers which are initially set to zero.

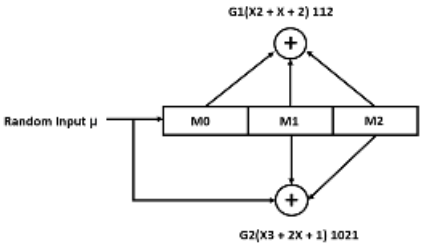


Fig. 1 Convolutional Encoder for code rate ½

Proposed design having two generator polynomials are G1=m0^2 +m1 +2m2, G2= u^3 +2m1 +m2. Convolutional encoder block diagram is shown in figure.2. It is having four memories elements.one is used to store the input bit and three are used to store the previous input bits. Memory element is designed using D Flip Flop. Generator polynomial is designed using simple adder or XOR gate. Depending on the polynomial equation encoder outputs the encoded data.

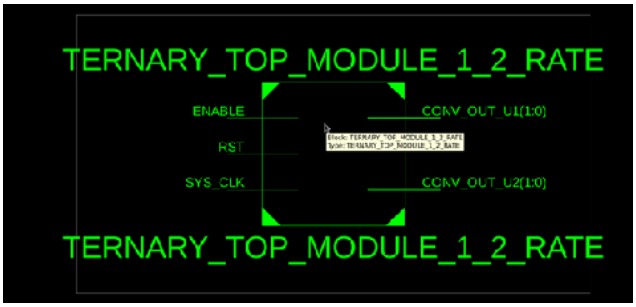


Fig. 2 Ternary top module for code rate ½

Shown in Fig. 3 is a (3, 1, 3) convolutional encoder. This encoder is going to be used to encode the 2 bit input sequence [00,01,10] with the three generator polynomials specified by the bits [1 2 1] and [0 1 0 ] and [2 0 1] u represents the input bit, and G1 and G2 and G3 represent the output bits 1 and 2 and 3 respectively. m0 and m1 and m2 represent the initial state of the memory registers which are initially set to zero.

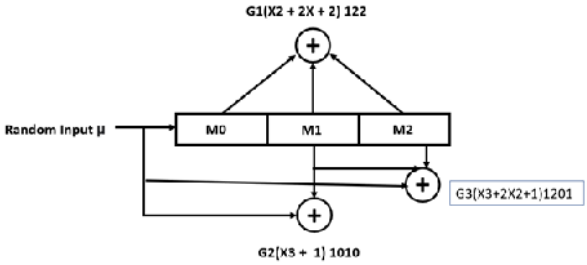


Fig 3 Convolutional Encoder for code rate 1/3

Code rate of convolutional encoder is 1/3. Generator polynomial are the mathematical representation of the convolution encoder. Proposed design having three generator polynomials are G1=m0 +2m1 +2m2, G2= u +2m1, and G3 =u+2m0+m2. Convolutional encoder block diagram is shown in figure.4. It is having four memories elements.one is used to store the input bit and three are used to store the previous input bits. Memory element is designed using D Flip Flop. Generator polynomial is designed using simple adder or XOR gate. Depending on the polynomial equation encoder outputs the encoded data.

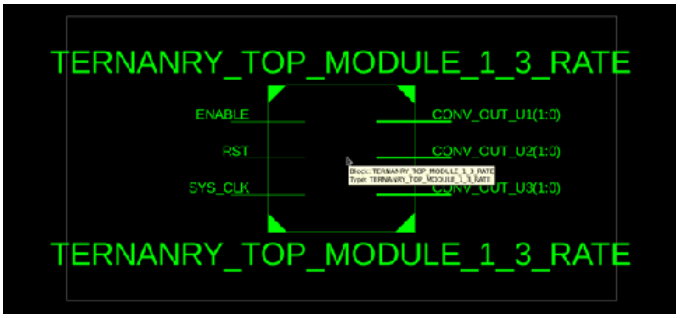


Fig 4 Ternary top module for code rate 1/3

The convolutional encoder for code rate2/3 and constraint length *K*=3 is represented in Figure 5. The convolutional encoder for *K*=3 has three shift registers namely Reg 1, Reg 2 , Reg 3 and two modulo-3 adders with the generator polynomial in the upper path being given by (1 0 2 1 ) and (1 2 0 1 ) and has two shift register and one modulo 3 adder with generator polynomial in the lower path being given by (101).The output sequence is obtained with the help of the generator polynomials and the message sequence, using the expression of generator polynomial.

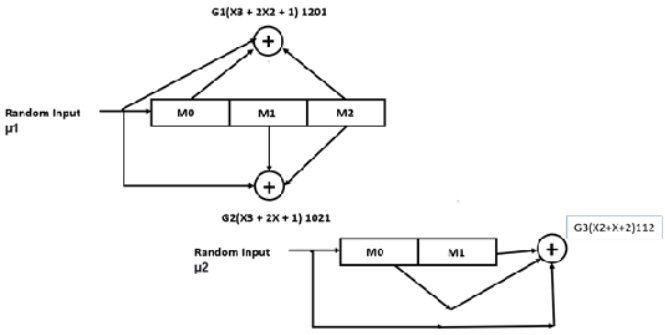


Fig 5 Convolutional Encoder for code rate 2/3

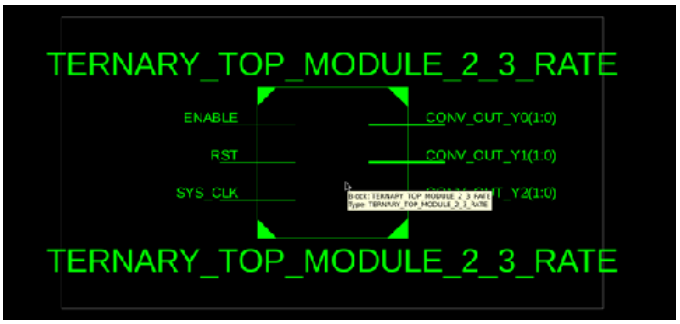


Fig. 6 Ternary top module for code rate 2/3

## III. DISCUSSION OF TERNARY CONVOLUTIONAL ENCODER

TABLE1 shows the state table of ½ convolutional encoder, which is consisting of current state, encoded data i.e., output of the convolutional encoder and shifted register value i.e., next state. Encoder will use the input bit with register values to produce output. In figure m0, m1, and m2 are the values stored in the register and G1 and G2 are the output values in table 1. The proposed design consisting of twenty-seven states values from 000 to 222. Here in table, we represent 5 combinations only out of 81. In this design considered zero is the initial states. For code rate ½ encoder, the combined output sequences of the two modulo-3 adders are given by {00 01 22 10 11 12 20 21 22}. For a sample input sequence: (012012012) and the present state shift register is obtained as: (000 100 200 010 110 210 020 120 220).

**Table 1** .State table for 1/2 code rate convolutional Encoder

| Present state(m0m1m2) | Input u | Next state (m0+m1+m2+) | Output(G1 G2) |
|---|---|---|---|
| 001 | 0 | 000 | 21 |
| 010 | 2 | 201 | 12 |
| 110 | 1 | 111 | 20 |
| 020 | 0 | 002 | 21 |
| 120 | 1 | 112 | 00 |

TABLE 2 shows the state table of 1/3 convolutional encoder, which is consisting of current state, encoded data i.e., output of the convolutional encoder and shifted register value i.e., next state. Encoder will use the input bit with register values to produce output. In figure m0, m1, and m2 are the values stored in the register and G1,G2 and G3 are the output values in table 2. For code rate 1/3 encoder, the combined output sequences of the three modulo-3 adders are given by {000 011 022 102 110 121 201 212 220}. For a sample input sequence: (012012012) and the present state shift register is obtained as: (000 100 200 010 110 210 020 120 220.

**Table 2 .**State table for 1/3 code rate convolutional Encoder

| Present state(m0m1m2) | Input u | Next state (m0+m1+m2+) | Output(G1 G2) |
|---|---|---|---|
| 001 | 0 | 000 | 201 |
| 010 | 2 | 201 | 202 |
| 110 | 1 | 111 | 020 |
| 020 | 0 | 002 | 120 |
| 120 | 1 | 112 | 200 |

The following table 3 shows the incoming bit U1 and U2 and status of encoder i.e. m0 ,m1 and m2 and corresponding output

Table 3 .State table for 2/3 code rate convolutional Encoder

| Present state(m0m1m2) | Input u1 | Next state (m0+m1+m2+) | Output(G1 G2) | Present state (m0m1) | Input u2 | Next state (m0+m1+) | Output G3 |
|---|---|---|---|---|---|---|---|
| 001 | 0 | 000 | 201 | 00 | 2 | 20 | 1 |
| 010 | 2 | 201 | 202 | 10 | 2 | 21 | 0 |
| 110 | 1 | 111 | 020 | 11 | 0 | 01 | 2 |
| 020 | 0 | 002 | 120 | 21 | 0 | 02 | 2 |
| 120 | 1 | 112 | 200 | 22 | 1 | 12 | 1 |

## IV. SIMULATION RESULT

Here the entire design of the Convolutional Encoder was implemented using VHDL programming language and the simulation were done and tested on the XILINX ISE 9.1 simulator. The key operation in the process of convolution is multiplication which is implemented using shift and addition .The addition operation dominates complexity in comparison to shift operation . and consume significant amount of dynamic power . while encoding .Hence optimization of adder utilization becomes a key factor while implementing reconfigurable hardware . A ternary convolutional encoder can be implemented using shift registers and modulo 3 adders. When starting coding, all these registers are reset for initialization. Every shift register is equivalent to a flip flop. These flip flops are connected in series to complete shifting and updating operation under the action of the clock pulse. In Figure , the squares are memory elements and the circled numbers are the coefficients that the contents of the memory elements are multiplied by. The circled sum represents a modulo-3 adder.

For ternary convolutional encoder the output V1 and V2 is depend upon the clock, reset and the data d from ternary randomizer . and the respective timing wave form of V1 is110001 and the output V2 is 100110 where the Reset signal is initiated at falling edge of the first clock signal.

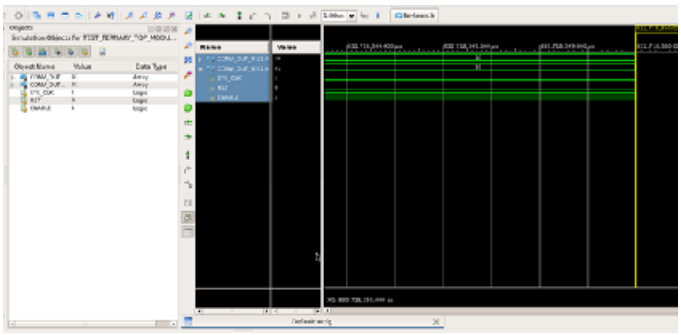

Fig 7 Simulation result for ½ Ternary convolutional Encoder

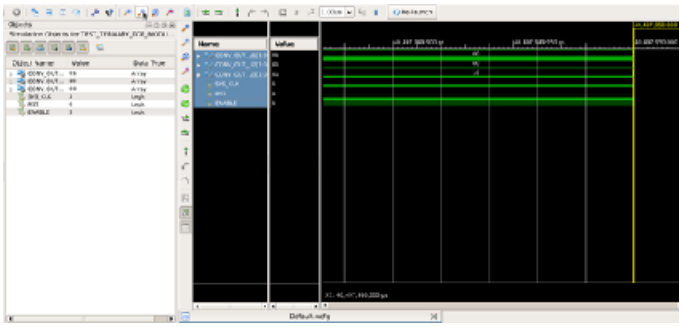

Fig 8 Simulation result for 1/3 Ternary convolutional Encoder

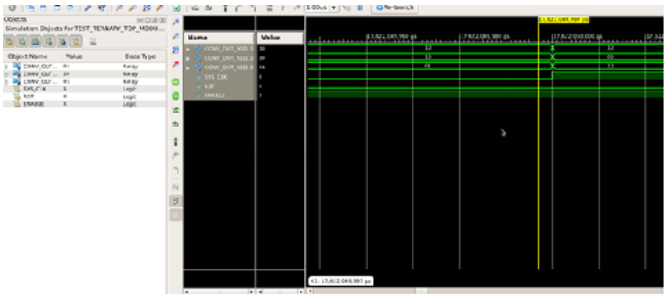

Fig 9 Simulation result for 2/3 Ternary convolutional Encoder

## V. CONCLUSION

The use of ternary systems can reduce the required hardware resources and is considered a promising technology for the future. Adaptive convolutional encoding algorithm on the other hand reduces computation time by dynamically adopting to reduce the iterations and computing resources. In the recent years, there has been an increasing demand for efficient and reliable digital data transmission and storage system. This demand has been accelerated by the emergence of the large-scale, high speed data networks for the exchange, processing, and storage of digital information in the Military, Government, and Private spheres. A merger (i.e., combination) of Communication and computer technology is required in the design of these systems.

## REFERENCES


[1]. Fazal Noorbasha, G.Jhansi , K.Deepthi , K Hari Kishore, "ASIC Implementation of Convolution Encoder and Viterbi Decoder Based Cryptography System" (IJITEE) ISSN: 2278-3075, Volume-8 Issue-6S, April 2019.

[2]. Zhang Taotao, Zhang JingKe, Zhou Zhiwen, Yang Zhifei, Liu Wanhong, "FPGA-Based Large Constraint Length Convolution Code Encoder Verification, ICSP 2019

[3]. Dr. S Aruna, Mr. A.V. Adiseshu, Dr.K.Srinivasa Naik , "Design of Viterbi Decoder for Speech to Text Conversion Application using ACS Architecture,IOSR journal of VLSI and signal processing Jan 21, 2019

[4]. Gabriele Meoni, "Design Optimization for High Throughput Recursive Systematic Convolutional Encoders" ICSTCC 2018

[5]. Prof. Vijaya Bharathi M, Sneha.H., Mahesh.M., Shwetha.N., Sowmya.S, "Forward Error Correcting Implementation Using Convolutional Encoders and Viterbi decoding" International Journal of Electrical, Electronics and Computer Systems (IJEECS) Volume -6, Issue-3, 2018

[6]. Mr.J.Anuj Sai, Mr.P.Kiran Kumar , "FPGA Design and Implementation of Convolution Encoder and Viterbi Decoder IJSER,Vol 9,issue 3, March 2018

[7]. Girish D. Korde1, Sanjay L. Haridas, "Design of Asynchronous Viterbi Decoder Using Pipeline Architecture" IJRASET Volume 6 Issue I, January 2018

[8]. Mr Sanket Kadu, "Design and Implementation of Viterbi Encoder and Decoder on FPGA" IJIRST –International Journal for Innovative Research in Science & Technology| Volume 3 | Issue 10 | March 2017

[9]. Akash Thakur and Manju K Chattopadhyay, "Design and Implementation of Viterbi Decoder Using VHDL, 3rd International Conference on Communication Systems (ICCS-2017)

[10]. Moussa Hamdan1 and Abdulati Abdullah2, "Analysis and Performance Evaluation of Convolutional Codes over Binary Symmetric Channel Using MATLAB" Proceedings of The Second International Conference on Electrical and Electronic Engineering, Telecommunication Engineering, and Mechatronics, Philippines 2016

[11]. Mahmoud Abdelaziz and T. Aaron Gulliver,"Ternary Convolutional Codes for Ternary Phase Shift Keying" IEEE COMMUNICATIONS LETTERS, VOL. 20, NO. 9, SEPTEMBER 2016

[12]. Juganpreet Kaur Brar, R K Bansal, Savina Bansal, "Goodness Analysis of Generator Polynomial for Convolution Code with Varying Constraint Length", IJARCCE, Vol 5, issue 11, Nov 2016

[13]. Gaurav Purohit,1 Kota Solomon Raju,2 and Vinod Kumar Chaubey1, "XOR-FREE Implementation of Convolutional Encoder for Reconfigurable Hardware, Hindawi Publishing Corporation International Journal of Reconfigurable Computing Volume 2016

[14]. Deepa Kumari*, Madan Lal Saini "Design and Performance Analysis of Convolutional Encoder and Viterbi Decoder for Various Generator Polynomials" IJERA Vol. 6, Issue 5, (Part - 2) May 2016

[15]. Do Duy Tana and Yeon-Mo Yangb , "Design and Simulation of Rate One-Third Convolutional Codes with Viterbi Algorithm based Hidden Markov Model for Digital Communications" Journal of Multidisciplinary Engineering Science and Technology (JMEST) ISSN: 2458-9403 Vol. 3 Issue 10, October – 2016

[16]. Arpitha K H1, Dr. P A Vijaya2, "Design of Low Power Efficient Viterbi Decoder" (IJRSEEE) Volume 2, Issue 2, 2016, PP 1-7

[17]. Khloud Mostafa, "High performance reconfigurable Viterbi Decoder design for multi-standard receiver", 33rd NRSC,Feb 2016

[18]. Rakhi B. Menon, Dr. Gnana Sheela K, "Synthesis of Convolution Encoder and Viterbi decoder of rate 2/3 using Xilinx ISE tool, IJARECE Vol 5, issue 2, Feb 2016

[19]. Hao Peng, Rongke Liu, Yi Hou and Ling Zhao, "A Gb/s Parallel Block-based Viterbi Decoder for Convolutional Codes on GPU", JULY 2016

[20]. Bhaskar Nandy, "Analysis of Convolutional Encoder System",IJCET (April 2016)