Reconfigurable Ternary Computing for Convolutional code

Mayuri Chetan Rathi , Dr C.N.Deshmukh*

Research Scholor,PRMIT& R Badnera,Amravati,444607,India
**Professor,PRMIT& R Badnera,Amravati,444607,India*

Abstract

The need for trustworthy and efficient digital data transmission and storage systems has increased in recent years. The demand to handle and store digital information has grown in the business sector, the military, and the government as a direct result of the widespread availability of high-speed, large-scale data networks. This requirement must be met in order for the design of these systems to keep up with the rapid pace set by communication and computer technologies. When trying to overcome the variable degradation in real time, one of the most important factors to consider is the dependability of the broadband communication channel. Therefore, the use of convolutional codes and other channel-coding strategies is an essential component of any broadband communication system. DSL, WLAN, and 3G standards all require different configurations of convolutional coding, each of which must achieve a specific level of coding performance despite operating at a unique data rate (constraint length and code rate) Therefore, from the perspective of channel-coding techniques, hardware implementations for the development of an encoder are essential. This encoder should be able to support multiple networks using a reconfigurability approach and should be able to function across a variety of standards. Additionally, flexibility and hardware performance should both be prioritised. This calls for forward error control coding with reconfigurable logic, which provides high-speed, low-power dynamically dedicated hardware architectures under a number of speed/power performance constraints at different time intervals that can function within a variety of channel conditions. The ternary computation system has a number of benefits, the most notable of which are the availability of a high data rate, improved spectral efficiency, enhanced coverage, and lower latency. As a result, there is a growing demand for ternary-based systems that use convolutional encoders as a result of developments in technology
Keywords: Ternary computing ,Reconfigurable logic ,Ternary communication .

1. Introduction

Within the parameters of wireless communication standards The goal of reconfigurable computing is to satiate the need for both fast data rates and low power usage. In the present day, we may utilise digital mobile phones, the internet, and CD/DVD players to transmit digital data from one location to another or retrieve data from a storage device. Sometimes it's necessary to ensure that data is sent without any interruptions. In order to keep these mechanisms running smoothly. Efficient channel coding methods are utilised for this purpose. The error-correcting abilities of a system are determined by the coding rate, generator polynomial, and constraint length. In order to ensure the integrity of data transfer in modern wireless standards and satellite communication, convolutional codes are utilised as a fundamental building component. This includes the transmission of digital video, radio, and mobile communication. The message is encoded into symbols by the transmitter and sent along the transmission channel. The data is encoded in many stages using a combination of hardware and software for processing.

1.1 Digital computations operations on the digits 0, 1, and 2 (the "ternary" numbers) are carried out. creating high-density, large-scale chips High-bandwidth data transmission in parallel and serial communication, as well as enhanced manufacturing density that presents hurdles for interconnections and pin-out issues, are two of binary logic's drawbacks. Multi valued logic is an appealing and a key field of study because of the benefits it offers, chief among them decreased complexity in the design, which leads to greater manufacturing density and high-bandwidth data connection enabling parallel and serial data transfer on a smaller on-chip. As a special case of Multivalued logic, Ternary logic distinguishes between three distinct values: true, false, and intermediate.

1.2 Description of invention

To begin modelling a convolutional encoder, one must first generate Ternary Random Data. After ternary random data was generated, the convolutional encoder's coding rate, constraint length, and number of shift registers were set to get the encoder's final state count. The zero-value starting point was used as the first setting for the encoder. The output's link to the shift registers is crucial to the Convolutional code encoder's design. The input bits are temporarily stored in the fixed length shift register. Each

* Dr C.N.Deshmukh
E-mail address:2014mmsoni@gmail.com

shift register's output is sent into a modulo 3 adder, which combines all of the results. In a convolutional code encoder, the most significant bit of a shift register is fed the message bits. It takes K=m+1 shifts for a message bit to go from the input to the output of an encoder with an m-stage shift register, and the encoder's memory is equal to the number of message bits. If you think of the Convolution encoder's shift register as a finite state machine, the set of delay elements inside has d potential states, where d is the number of delay elements in the register. based on the current state of affairs and the information at hand. In this case, a codeword is created from bits. These calculations are only the sums of predetermined shift register tap sequences. The history of the message bits and the current message word are stored in a register, the status of which affects each codeword. Consequently, the meanings of consecutive codewords rely on the meanings of the preceding ones. The efficiency of the convolution code is determined by the connections between the shift registers and the mod-3 adder. In this work, the convolutional encoder structure is shown for various code rates, taking into account the coupling of shift registers with output through a mod-3 adder. Convolutional codes, in contrast to block codes, don't have a fixed word length and may instead have their properties altered by changing the connections between nodes. In convolution, the multiplication action is realised by shifts and sums. When encoding, the addition operation requires a lot of dynamic energy. And so, while implementing using reconfigurable hardware, which is more difficult than shift operations. optimising adder use has paramount importance.

2. 2.Illustrations of Ternary Convolutional Encoder

Convolutional codes are often used as the coding method of choice in practical communication systems. Ones who can compress a large amount of information into a single word. The decoding of a symbol depends on both the current symbol and the one that came before it in the input stream. Different factors including coding rate R, constraint or memory length m, and free distance dfree are taken into account. We choose the best convolutional code for each application.

Figure 1 shows a convolutional encoder with a coding rate of 2/3 and a constraint length of K=3. The convolutional encoder for K=3 consists of three shift registers (Reg 1, Reg 2, and Reg 3) and two modulo-3 adders, with the generator polynomial in the upper path given by (1 0 2 1) and (1 2 0 1), and two shift registers and one modulo-3 adder, with the generator polynomial in the lower path given by (2 0 0 1). (101). The output sequence is calculated by using the message sequence and the generator polynomials.

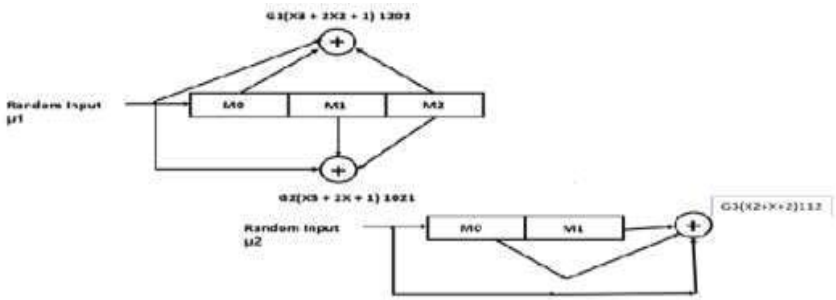

Fig. 1. Convolutional Encoder for code rate 2/3

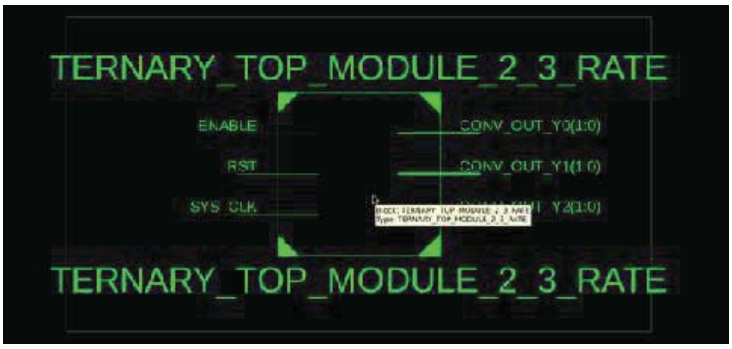

Fig. 2. RTL Schematic Of Convolutional Encoder for code rate 2/3

The generator polynomials are given by *gi(1)* = { $x1^3+2x1^2+1$} and *gi(2)* = { $x1^3+2x1+1$}and *gi(3)={* $x2^2+x2+2$} using the expression of generator polynomial .
It is assumed that the message sequence m = (00101101) is completely at random, and the resulting sequence is calculated as

follows. determined by the initial polynomial of the generator The calculated result is as follows: The XOR operation is accomplished by multiplying the first bit of the generator polynomial gi (1) by 1, the second bit by 1, and the third bit by 2. The first result from the higher modulo-3 adder When calculating the second output of a modulo-3 adder, the first bit of the message sequence (012012) is multiplied by 1, the second bit of the generator polynomial gi (1) is multiplied by 2, and the third bit is multiplied by 1. The output is found at the lower modulo-3 adder once the following computation is made. Upper modulo-3 adder O/P1 yields the following sequence of outputs: (012201). Upper modulo-3 adder O/P2 yields the following output sequence: (012012). Generating lower modulo-3 adder outputs with gi(3). The output of the lower modulo-3 adder is calculated by multiplying the first bit of the generating polynomial gi(3) by the first bit of the message, and then multiplying the second bit by factor 2. Sequence of results from the lower modulo-3 adder O/P2 is (00100110). V1 and V2 are updated to reflect the current conditions. m0,m1,m2 . The bits m0 and m2 hold the prior two messages. The left-to-right ', movement of the message bit. There is a current bit in m0.

The output switch first samples the input V1, and then it samples the input V2. Through the shift register, we may transfer the data from memory location m1 to memory location m2 and from memory location m0 to memory location m1. Then, we take the following input and store it in the variable m0. Based on this most recent combination of initial circumstances, m1 and m2 are generated once again. At that point, V1 and V2 are measured by the output switch. Each bit of the input message is represented by two encoded bits, V1 and V2. To encode a message using a convolutional encoder, you'll need three bits for a code word and two for the message.

For the top route, k=1 denotes the number of message bits and n=2 denotes the number of encoded output bits for a single message bit. For the lowest possible route, the number of message bits is k=1, and the number of encoded output bits for a single message bit is n=1.

3. DISCUSSION OF TERNARY CONVOLUTIONAL ENCODER

Table1 displays the stages of a 2/3 coding rate convolutional encoder. The current observed state, the encoded data generated by the convolutional encoder, and the value moved into the register to represent the next observed state are all recorded here. The following table displays the input bits (U1 and U2), the current states (m0, m1, and m2) of the encoder, and the related output. Given that there are three trit streams heading out (v1, v2, and v3) and only two leading in (u1, u2), the coding rate is 2/3. S0, S1, S2, and S3 are only three of the twenty-seven potential states. Upto S26 . The current state will transition to the next stage if the input value changes from 0 to 1 or 2.

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

4.RECONFIGURABLE TERNARY CONVOLUTIONAL ENCODER

A convolutional encoder is a finite state machine that, given an input sequence u of information blocks uj, u =... u2u1u0u1u2, performs a linear mapping, where each block uj contains k symbols, that is, uj = (u(1) j, u(2j,..., u(k)j), where u(i) The convolutional encoder receives this information stream in blocks. The encoder is given the k-ary block uj at time j, and it spits out the n-ary block vj. If uj is the "information block delivered at time j," then j is the "time index" of uj. The jth data block uj has a matching code block denoted by the letter vj. To express the number of symbols in each code block, vj, we write vj = (v(1) j, v(2) j,..., v(n) j), where v(i) j 2 R, for I = 1,..., n. To do this, it generates the output sequence v =... v2v1v0v1v2... which is treated as a single codeword by the encoder. Both u and v must begin at a certain moment and might potentially conclude at the same point. The most notable feature of a convolutional encoder is that the j-th code block vj is not only dependent on the current uj but also on the previous, say, fixed number of information blocks uj1, uj2,. This may be written as vj = ujG0 + uj1G1 +... + ujG. The inputs and outputs produced by this flexible top block are governed by the coding rate of the convolutional encoder. The logic block seen in Fig. 2 may be set up in many ways to accommodate any desired convolutional encoder code rate.

.

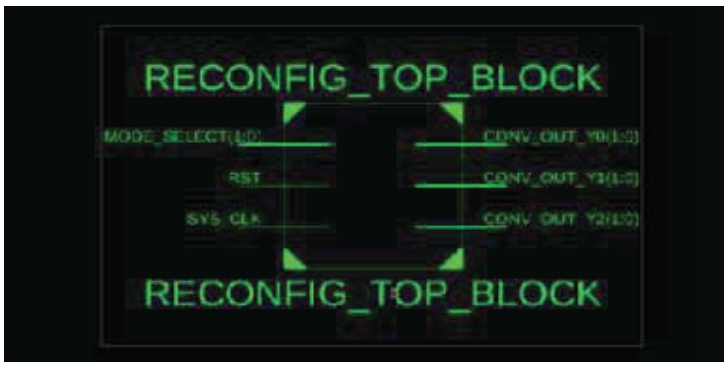

Fig. 2. Reconfigurable Convolutional Encoder for code rate ½ ,1/3,2/3

Since the data rate, bandwidth, and performance characteristics of the physical layer are defined by emerging wireless communication standards, the physical layer must dynamically adapt to accommodate the frequent changes in these standards. One of the main stumbling blocks to a dependable and high-throughput wireless network is channel faults. The physical layer's crucial block, consisting of convolutional encoders and Viterbi decoders, must be built to accommodate fluctuations in data rate and channel noise by running at high frequency on a low power supply and with the capacity to be reconfigured.

5. Simulation Result

Figure is a timing diagram for a Ternary Randomizer. In this case, ISE was used for every step of the Convolutional Encoder's design and implementation. Xilinx ISE 14.7 Simulator is used for simulation and testing, and it requires a virtual machine setup and the Verilog programming language. When creating a random ternary bit stream, it is common practise to first employ a pseudo random ternary sequence as a template. For the generation of this ternary bit stream, a linear feedback shift register might be employed (LFSR). Ternary convolution encoding receives data from the Ternary Randomzier. Convolution relies heavily on multiplication, thus we employ shift and addition to do the necessary multiplication operations. Therefore, it is crucial to optimise adder utilisation throughout the construction of reconfigurable hardware. At the outset of encoding, the convolutional encoder's linear feedback shift register is reset, so initialising all of its registers. As a generalisation, you may think of shift registers as "flip flops on steroids." These flip flops are chained together in sequential order to perform the updating and switching triggered by the clock pulse. When drawing a convolutional encoder diagram, each square represents a memory element, and the ringed numbers are the coefficients used to multiply the information in those memory cells. The circular sum represents a modulo-3 adder. The encoder's output may take on any value in the ternary number system, and the reset input makes it happen.

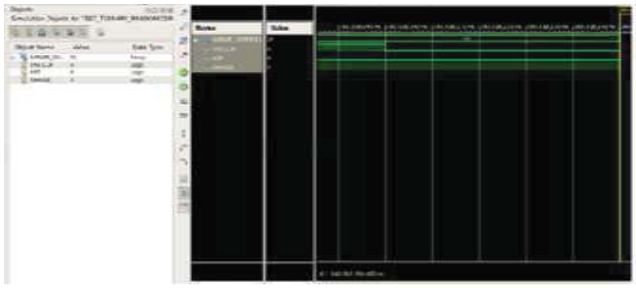

Fig. 3. Simulation result for Ternary Randomizer

This encoding scheme is a ternary convolutional encoder. Ternary randomizer outputs V1, V2, and V3 depending on clk, rst, and data d, as seen in the figure above; V1's timing wave form is 110001, V2's is 100110, and V3's is 011011; the Reset signal begins at the falling edge of the first clock signal. Convolution Encoder and Viterbi Decoder for xc3s250e-4-pq208 Spartan 3E board with constraint length 3 and coding rate 2/3 are shown here. The Hardware Description Language is used to document the design. The FPGA is independently functional as a Convolutional encoder over a range of coding rates, thanks to simulation and

synthesis in Xilinx ISE 14. By setting it up in this fashion. Therefore, it is advantageous to employ ternary logic since it allows for the use of the same hardware.

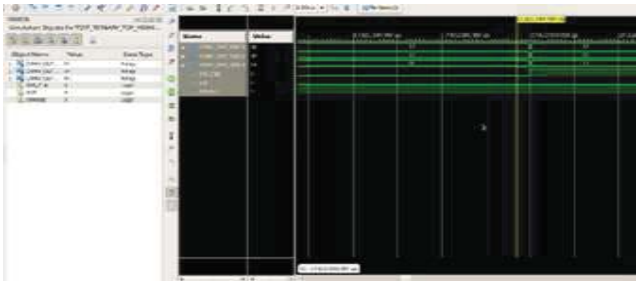

Fig. 4. Simulation result for Ternary Convolutional Encoder for code rate 2/3

Fig. 5. Simulation result for Reconfigurable Convolutional Encoder

With regards to processing speed, wiring complexity, energy consumption, and wireless communication networks, ternary systems outperform their binary counterparts hands down. where. There are several benefits to using a ternary system instead of a binary one. Among the many benefits of the Ternary system are: As the amount of connections used to implement logic functions is decreased, a lot of space is freed up within the ship. Because more data may be sent across the same number of lines, less storage space is required for the same amount of data. This permits serial and certain serial-parallel operations to be carried out more quickly. On the other hand, the adaptive convolutional encoding technique constantly adapts to minimise computation time, iterations, and resources. Convolution codes, which are optimised for greater free distance, are preferred because of their error correcting capabilities, which are connected to polynomial strength. Typically, shift register (SR) is used to implement the convolutional encoder, which consists of delay components and modulo-3 adders (XOR gates). Convolution relies on multiplication, which is realised by shifts and adds. A lot more dynamic power is used up during encoding due to the addition operation, which is also the most difficult compared to the shift operations. Implementing with reconfigurable hardware necessitates, thus, careful consideration of adder use optimization. Reconfiguration is possible with little effort by simply altering a polynomial with the same constraint length and coding rate in a hardware implementation using ROM. Overcoming the dynamic decline in dependability of a broadband communication connection in real time is a crucial challenge. Consequently, channel-coding methods, such as convolutional codes, are crucial components of any broadband communication system.

A common place reconfigurable convolution encoder consists mostly of a shift register with N stages and v modulo-3 adders. A wide variety of modulo-3 adders may be constructed using XOR gates alone. A convolution encoder is defined by the parameters (N, k, v). In this notation, v represents the total number of encoder outputs, k represents the total number of encoder inputs, N1 represents the total number of memory components (Flip- Flops), and N represents the total length of the constraints. The k/v notation represents the encoding speed of the corresponding encoder. Into the encoder's N-stage shift register goes the source data sequence represented by Input(n) = (input(0), input(1), input(2),...). Using modulo 3 adders, we transform each input into a distinct series of output values (V1, V2). Because of the importance of the constraint length to convolution encoder performance, a better understanding of it is necessary. Keeping the constraint length and code rate the same while changing the generator polynomial via architectural reconfigurability might streamline the design.

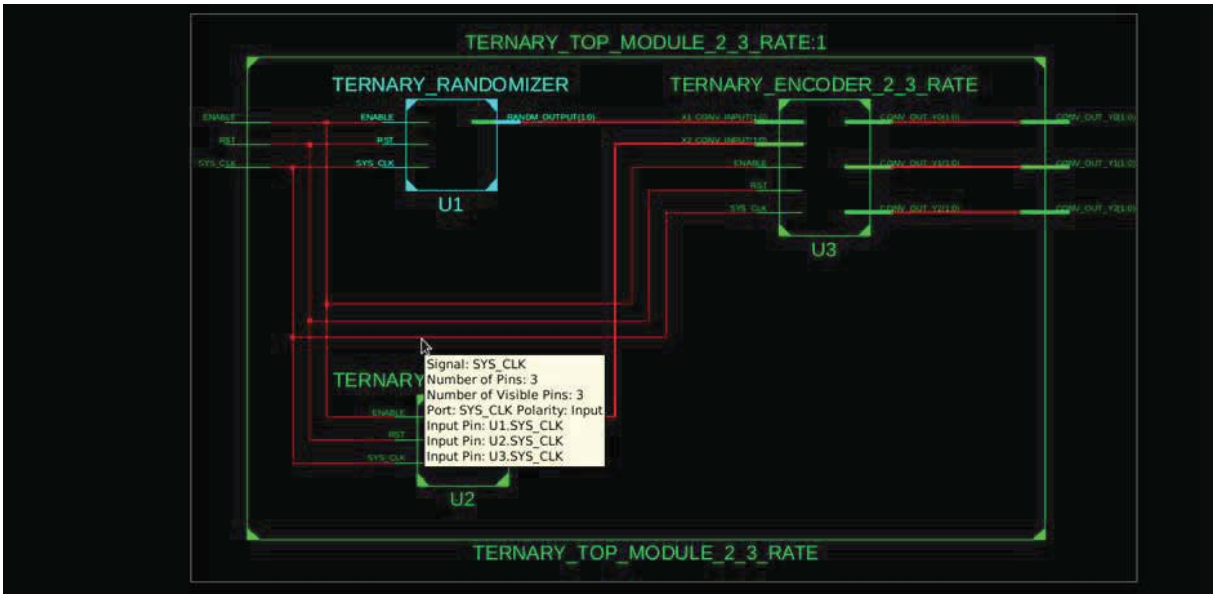

Fig. 7. Ternary Top module for1/2, 1/3 and 2/3 code rate With Reconfigurable Architecture

6. CONCLUSION

The capacity to reconfigure is crucial for satisfying the rising physical layer requirements of wireless communications, which call for faster data rates with lower power consumption. For long constraint lengths, the design and implementation of such structures presents a significant challenge. However, applications needing codes with enormous constraint lengths, such as 3G, generally render these designs impractical due to the hardware cost of reconfigurable logic of the convolutional codes increasing exponentially with the constraint length. Today, wireless connections are an integral part of most communication networks. Data services, such as websites and multimedia files, often use such connections to send and receive data, but they have varying needs in terms of bandwidth, latency, and reliability. The increasing demand for these offerings motivates researchers to look for ways to improve efficiency without increasing complexity. One way to do this is by use ternary symbols. Because of their potential to reduce the amount of hardware needed, systems based on ternary logic are viewed as a promising future technology. New requirements for channel coding arise with the advent of broadband Internet access. Using parallel and pipeline characteristics of the hardware resources, the architecture is built on programmable logic arrays. Through the use of HDL language, simulation, synthesis, and implementation are performed on the existing algorithm with the help of EDA tools based on FPGAs. The system as a whole now operates more efficiently in terms of both space and time.

Acknowledgements

Author 1: Mayuri Rathi is currently working as an Assistant Professor in Modern Education Society College of Engineering, Wadia campus, Pune. She is MTech in VLSI design and having teaching experience of 8 years. She is a pursuing PhD under the guidance of Dr C.N. Deshmukh. Her area of interest is Radio frequency communication, VLSI Design, Antenna Design.
Author 2: Dr C.N.Deshmukh is currently working as a Associate Professor in Prof Ram Megha Institute Of Technology and Research ,Badnera ,Amravati .He is having teaching experience of 30 years. He has guided many students for their research work.currently five research scholars are pursuing their PhD under his guidance. His area of interest are wireless communication, Digital Image Processing ,Ternary Computing .

References

1. S. Madumitha and G. Suchitra, “performance analysis of a convolutional encoder using digital modulation technique,” ictact journal on communication technology, march 2020, volume: 11, issue: 01
2. 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.

3. Zhang Taotao, Zhang JingKe, Zhou Zhiwen, Yang Zhifei, Liu Wanhong, "FPGA-Based Large Constraint Length Convolution Code Encoder Verification, ICSP 2019
4. 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
5. Gabriele Meoni, "Design Optimization for High Throughput Recursive Systematic Convolutional Encoders" ICSTCC 2018
6. 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
7. 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
8. Girish D. Korde1, Sanjay L. Haridas, "Design of Asynchronous Viterbi Decoder Using Pipeline Architecture" IJRASET Volume 6 Issue I, January 2018
9. 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
10. Akash Thakur and Manju K Chattopadhyay, "Design and Implementation of Viterbi Decoder Using VHDL, 3rd International Conference on Communication Systems (ICCS-2017)
11. 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
12. Mahmoud Abdelaziz and T. Aaron Gulliver,"Ternary Convolutional Codes for Ternary Phase Shift Keying" IEEE COMMUNICATIONS LETTERS, VOL. 20, NO. 9, SEPTEMBER 2016
13. 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
14. 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
15. 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
16. 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
17. Arpitha K H1, Dr. P A Vijaya2, "Design of Low Power Efficient Viterbi Decoder" (IJRSEEE) Volume 2, Issue 2, 2016, PP 1-7
18. Khloud Mostafa, "High performance reconfigurable Viterbi Decoder design for multi-standard receiver", 33rd NRSC,Feb 2016
19. 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
20. Hao Peng, Rongke Liu, Yi Hou and Ling Zhao, "A Gb/s Parallel Block-based Viterbi Decoder for Convolutional Codes on GPU", JULY 2016
21. Bhaskar Nandy, "Analysis of Convolutional Encoder System",IJCET (April 2016)
22. Rakhi B. Menon, 2 Dr. Gnana Sheela, "Design and Implementation of Convolution Encoder and Viterbi Decoder,IJCSN,volume 4 ,issue 6, December 2015
23. Ranjitha S, Divya Preethi K, Megha K,, "An Efficient FPGA Implementation of Convolutional Encoder and Viterbi Decoder for DSP Applications", International Journal of Engineering Research & Technology (IJERT Vol. 4 Issue 10, October-2015
24. Gaurav Vijay, R.P Gupta, "Performance Analysis of Convolutional Codes with Bit Error Rate V/S SNR over Channel for Viterbi Algorithm"IJIRCCE ,Vol 3 ,issue 6, June 15
25. D. Vaithiyanathan a,, J. Nargis b, R. Seshasayanan n, "High performance ACS for Viterbi decoder using pipeline T-Algorithm, ELSEVIER 15 April 2015
26. NEHA, GH.MOHAMMAD RATHER, "CONVOLUTION ERROR CONTROL CODING -A REVIEW", International Journal of Electrical, Electronics and Data Communication, ISSN: 2320-2084 Volume-3, Issue-3, March-2015