

**SDR IMPLEMENTATION OF CONVOLUTIONAL ENCODER  
AND  
VITERBI DECODER USING QPSK MODULATION TECHNIQUE**

A thesis submitted in partial fulfillment of the requirements

for the award of degree of

**MASTER OF TECHNOLOGY**

**In**

**VLSI Design and CAD**

Submitted By

**Abhishek Aggarwal**

Roll No. 601261001

Under the guidance of

**Dr. Rajesh Khanna**

**Professor, ECED**



**Department of Electronics and Communication Engineering**

**THAPAR UNIVERSITY, PATIALA**

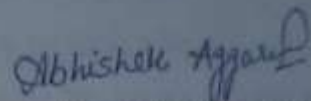
**July 2014**

## CERTIFICATE

I hereby declare that the work which is being presented in the thesis entitled, "SDR implementation of Convolutional Encoder and Viterbi Decoder using QPSK Modulation Technique" by me in partial fulfillment of the requirement for the award of degree of M.Tech in VLSI Design & CAD submitted in Electronics and Communication Engineering Department of Thapar University, Patiala is an authentic record of my own work carried out under the supervision of Dr. Rajesh Khanna, Professor, ECED.

The matter presented in this thesis has not been submitted in any other University/Institute for the award of degree.

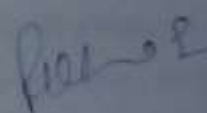
Date: 01 July, 2014

  
(ABHISHEK AGGARWAL)

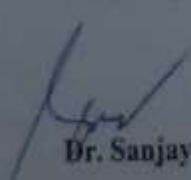
ROLL NO. 601261001

It is certified that the above statement made by the student is correct to the best of my knowledge and belief.

Date: 01 July, 2014

  
(Dr. Rajesh Khanna)  
Professor  
ECED, Thapar University

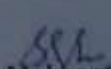
Countersigned By:-

  
Dr. Sanjay Sharma

Professor & Head

ECED, Thapar University

Patiala, 147004

  
Dr. S.K. Mahapatra

Dean of Academic Affairs

Thapar University

Patiala, 147004

## **ACKNOWLEDGEMENT**

To discover, analyze and to present something new is to venture on an untrodden path towards and unexplored destination is an arduous adventure unless one gets a true torch bearer to show the way. I would have never succeeded in completing my task without the cooperation, encouragement and help provided to me by various people. Words are often too less to reveals one's deep regards. I take this opportunity to express my profound sense of gratitude and respect to all those who helped me through the duration of this thesis. I acknowledge with gratitude and humility my indebtedness to **Dr. Rajesh Khanna, Professor**, Electronics and Communication Engineering Department, Thapar University, Patiala, under whose guidance I had the privilege to complete this thesis. I wish to express my deep gratitude towards him for providing individual guidance and support throughout the thesis work.

I convey my sincere thanks to **Head of the Department, Dr. Sanjay Sharma** as well as **PG Coordinator, Dr. Kulbir Singh, Assistant Professor**, Electronics and Communication Engineering Department, entire faculty and staff of Electronics and Communication Engineering Department for their encouragement and cooperation.

My greatest thanks are to all who wished me success especially my parents. Above all I render my gratitude to the Almighty who bestowed self-confidence, ability and strength in me to complete this work for not letting me down at the time of crisis and showing me the silver lining in the dark clouds. I do not find enough words with which I can express my feelings of thanks to my dear friends for their help, inspiration and moral support which went a long way in successful competition of the present study.

**(Abhishek Aggarwal)**

## **ABSTRACT**

In today's era, efficient transmission is very important. In case of long distance transmission wired approach is adopted. But in wireless approach, efficient transmission as well as efficient reception is required. Moreover, the channel through which the communication is taking place has to be considered efficiently. The transmitted information will get modulated according to the various modulation techniques. Convolutional encoder is one of the techniques, which is used to correct erroneous bits at the receiver end. This technique is also called as forward error correction technique. At the receiver end to decode the convolutional codes, Viterbi decoding technique is used. The Viterbi decoder Algorithm is widely used for estimating and detecting problems in signal processing and digital communications. This algorithm is used to detect signals in communications channels with memory, and to decode the sequential error control codes which results in the enhancement of the performance of digital communication systems. The applications of the Viterbi decoding algorithm are: digital TV (QAM, ATSC, and DVB-T), satellite communications and radio relay.

In this thesis, a complete convolutional encoded communication system in AWGN channel is designed using VHDL code. The complete RTL schematic of convolution encoder, AWGN channel and Viterbi decoder are designed and presented in the thesis. The complete encoded system is tested for QPSK modulation technique and the results are plotted in terms of BER. The complete system can be further implemented in SDR. The timing summary after analysis got is; minimum period is 27.362ns (i.e. maximum frequency is 36.547 MHz). Maximum input arrival time before clock is 4.213ns and maximum output required time after clock is 4.182ns.

# TABLE OF CONTENTS

<i>Certificate</i>	<i>i</i>
<i>Acknowledgement</i>	<i>ii</i>
<i>Abstract</i>	<i>iii</i>
<i>Table of contents</i>	<i>iv</i>
<i>List of figures</i>	<i>vi</i>
<i>List of tables</i>	<i>vii</i>
<b>1. BACKGROUND</b>	<b>1</b>
1.1 Introduction	1
1.2 Purpose	2
1.3 Objectives	3
1.4 Report Structure	3
<b>2. LITERATURE REVIEW</b>	<b>4</b>
2.1 Conclusion	9
<b>3. CONVOLUTIONAL ENCODER AND VITERBI DECODER</b>	<b>11</b>
3.1 Coding	11
3.1.1 Block Coding	12
3.1.2 Convolutional Encoder	12
3.2 Convolutional Encoder	12
3.2.1 Encoder Parameters	12
3.2.2 Encoder Structure	13
3.2.3 Encoder States	15
3.3 Special case Convolutional Encoder	15
3.3.1 Punctured Codes	15
3.3.2 Dual K Codes	16
3.3.3 Systematic v/s Non-Systematic Codes	17
3.4 Encoder Operation	18
3.5 Convolutional Representation	22
3.5.1 State Diagram Representation	22

3.5.2 Tree Diagram Representation	24
3.5.3 Trellis Diagram Representation	24
3.6 Decoding	25
3.6.1 Sequential Decoding	25
3.6.2 Viterbi Decoding	26
3.7 Hard Decision Viterbi Decoding	26
3.7.1 Hamming Distance	26
3.7.2 Trellis Explanation	27
3.8 Soft Decision Viterbi Decoding	32
3.8.1 Minimum Euclidean Distance	32
3.8.2 Trellis Explanation	32
<b>4. VHDL AND SDR IMPLEMENTATION OF COMMUNICATION SYSTEM</b>	<b>38</b>
4.1 LFSR	38
4.2 Convolutional encoder	39
4.3 QPSK Modulator	40
4.4 AWGN channel	40
4.5 Viterbi Decoder	41
4.6 Final Proposed Design	42
<b>5. CONCLUSION AND FUTURE SCOPE</b>	<b>47</b>
9.1 Conclusion	47
9.2 Future scope	47
<b>6. REFERENCES</b>	<b>48</b>
List of Publications	51

## LIST OF FIGURES

3.1	Communication system with channel encoding/decoding operations	12
3.2	Convolutional encoder with $k = 2$ , $r = \frac{1}{2}$	14
3.3	Convolutional encoder with $k = 3$ , $r = \frac{1}{2}$	14
3.4	Convolutional encoder with $k = 5$ , $r = \frac{1}{2}$	15
3.5	Convolutional encoder with $k = 2$ , $r = \frac{1}{2}$ to explain encoder states	15
3.6	2/3 Puncture Coding	16
3.7	Dual- $k$ Convolutional Code	17
3.8	(4,3,3) Systematic Convolutional Code	18
3.9	Convolutional encoder with $k = 2$ , $r = \frac{1}{2}$ to explain encoder operation	18
3.10	Modulo Adder	19
3.11	Convolutional encoder example with 1 <sup>st</sup> bit	19
3.12	Convolutional encoder example with 2 <sup>nd</sup> bit	20
3.13	Convolutional encoder example with 3 <sup>rd</sup> bit	20
3.14	Convolutional encoder example with 4 <sup>th</sup> bit	21
3.15	Continued example of convolutional encoder	21
3.16	Convolutional encoder state diagram	23
3.17	Code tree representation 1	24
3.18	Code tree representation 2	24
3.19	Trellis diagram for (2,1,3) convolutional encoder	25
3.20	Block diagram of Viterbi decoder	26
3.21	Trellis and path metrics at the time $t = 0$	27
3.22	Trellis and path metrics at the time $t = 1$	28
3.23	Step 3(a) Trellis and path metrics at the time $t = 2$	28
3.24	Step 3(b) Trellis and path metrics at the time $t = 2$	29
3.25	Step 4(a) Trellis and path metrics at the time $t = 3$	29
3.26	Step 4(b) Trellis and path metrics at the time $t = 3$	30
3.27	Step 5(a) Trellis and path metrics at the time $t = 4$	30
3.28	Step 5(b) Trellis and path metrics at the time $t = 5$	31
3.28	Decoded sequence for noisy channel	31
3.29	Final output	32
3.30	Trellis and path metrics at the time $t = 0$	33

3.31	Trellis and path metrics at the time $t = 1$	33
3.32	Step 3(a) Trellis and path metrics at the time $t = 2$	34
3.33	Step 3(b) Trellis and path metrics at the time $t = 2$	34
3.34	Step 4(a) Trellis and path metrics at the time $t = 3$	35
3.35	Step 4(b) Trellis and path metrics at the time $t = 3$	35
3.36	Step 5(a) Trellis and path metrics at the time $t = 4$	36
3.37	Step 5(b) Trellis and path metrics at the time $t = 4$	36
3.38	Final output	37
	<b>PROPOSED DESIGN</b>	<b>38</b>
4.1	LFSR Implementation generating random sequence	38
4.2	Convolutional encoder	39
4.3	Schematic diagram of proposed encoder	39
4.4	Schematic of QPSK modulation with convolutional encoder	40
4.5	Example of an AWGN channel	41
4.6	Implementation of an AWGN channel	41
4.7	Schematic of Viterbi Decoder	41
4.8	Schematic of final proposed design	42
4.9	Graph of BER when $\alpha = 0$	42
4.10	Graph of BER when $\alpha = 1$	43
4.11	Graph of BER when $\alpha = 2$	43
4.12	Comparison among various roll-off factors	44
4.13	Device utilization summary of proposed design	44
4.14	Final synthesis report	45
4.15	Final output	45
4.16	SDR implementation of convolutional encoder and Viterbi decoder	46

## **LIST OF TABLES**

2.1	History of Convolutional encoder and Viterbi Decoder	4
2.2	Review of Previous Architectures	9
3.1	Parameter range for convolutional encoder	13
3.2	Convolutional encoder state table representation	22

## 1.1 Introduction

With the advancement in technology, people are looking for the simpler methods and cost effective technique for communication. SDR (software defined radio) is a technique which not only brings the flexibility but also it is cost effective and have power to drive communications forward with wide-reaching benefits realized by service providers and product developers through to end users[1].

Software defined radio (SDR) forum in collaboration with IEEE P1900.1 group, has given a clear overview of SDR technology and its benefits. Software Defined Radio is defined as: "*Radio in which some or all of the physical layer functions are software defined*"[1].

In SDR all of the radio's operating functions are implemented with the help of modifiable firmware operating on programmable processing technologies. These devices cover FPGA (field programmable gate arrays), DSP (digital signal processing), GPP (general purpose processors), SoC (programmable System on Chip) or many other programmable processors [1].

This thesis describes the design and implementation of Convolutional encoded communication system based on a FPGA based platform. Detailed implementation results (performance, code size, and FPGA resources utilization) are presented. The main goal of the design case presented is to provide insight into the design aspects of a complex system based on FPGAs [2]. The results prove that an implementation based FPGAs are adequate for the communication system where the expected volumes are rather small.

Convolutional encoding is a forward error correction technique that is used for correction of errors at the receiver end. Viterbi decoding is the technique for decoding the convolutional codes [3]. The Viterbi Algorithm, an application of dynamic programming, is widely used for estimation and detection problems in digital communications and signal processing. The thesis aimed at the real time implementation of convolutional encoder and Viterbi decoder over SDR using QPSK digital modulation techniques.

## 1.2 Purpose

There has been explosive growth in wireless communication in the past few decades.

First and Second generation	Voice communication
Third generation	CDMA and strong focus on wireless data services

As we move onto the third generation, the reality is that first generation 3G systems did not satisfy the customer requirement of high-speed transmission, and the rates supported in practice were much lower than that claimed in the standards [4]. Enhanced 3G systems were subsequently deployed to remove the deficiencies. But, the data rate capabilities and network architecture of these systems are insufficient to address the unsatisfied consumer. With these considerations in hand, the move to 4G technologies like 3G LTE (long term evolution) and WiMAX is proceeding at an extremely rapid rate. The goal of next generation systems is to provide high-data rate, low-latency, high reliability (minimize outages and connection drops) employing packet-optimized radio access technology supporting flexible bandwidth allocation. Additional key objectives are to drive down the cost of infrastructure equipment and consumer terminals and to employ a more efficient modulation scheme than the CDMA technology used in 3G systems in order to make more optimal use of precious communication bandwidth [5].

To meet all of these requirements a significant re-structuring of both the physical layer (PHY) and network architecture is required. The initial intent of this thesis is to generate the random bits for finite time period. Secondly, to design convolutional encoder and to generate an AWGN noise channel, so that the communication takes place through this medium. Thirdly, to get the RRC filter, used for the modulation purpose. Fourthly, combine the modulator, channel and Viterbi decoder and demodulator part together to get the original message at the receiver end.

Convolutional encoding is an FEC technique that is particularly suited to a channel in which the transmitted signal is corrupted mainly by additive white

Gaussian noise (AWGN). Viterbi algorithm is a well-known Maximum-likelihood algorithm for decoding of Convolutional codes. They have rather good correcting capability and perform well even on very noisy channels. It has been widely deployed in many wireless communication systems to improve the limited capacity of the communication channels [6]. Field programmable gate arrays (FPGAs) with their inherently parallel structure, are increasingly the technology of choice for addressing the compute and flexibility requirements of next generation systems. FPGA advantage for its friendly VHDL language interface and was chosen for RTL implementation of encoder and decoder.

The main Objective of this thesis is to describe performance analysis of FPGA based for proposed design and testing of Viterbi encoder and Decoder.

### **1.3 Objectives**

The main objectives of the thesis are

- (i) Realization random bit generator, convolutional encoder, AWGN channel, Viterbi decoder in VHDL
- (ii) Design of RRC filter in VHDL.
- (iii) Hardware realization of complete communication system in FPGA
- (iv) Testing of the proposed Viterbi encoder and decoder hardware architecture.

### **1.4 Report Structure**

In the following chapters,

Chapter-2 An exhaustive study over convolutional encoder and Viterbi decoder in the form of literature review is provided.

Chapter-3 An introduction to convolutional encoder and Viterbi decoder is given.

Chapter-4 Real time Implementation of convolutional encoder and Viterbi decoder is given.

## 2. LITERATURE REVIEW

An exhaustive survey has been carried out in context with the titled work to find out the current work.

Table 2.1 History of Convolutional Encoder and Viterbi Decoder

YEAR	WORK
1955	Convolutional codes were introduced by Elias as an alternative to Block Codes.
1957	Wozencraft proposed Sequential decoding as an efficient decoding scheme for convolutional codes.
1963	Massey proposed a less efficient but simpler-to-implement decoding method called threshold decoding.
1967	Viterbi proposed a Maximum Likelihood decoding scheme that was relatively easy to implement for coder with small memory codes.

Abstracts with their results and discussions of some of the most relevant papers are reported in the following paragraphs:

**Rohan M. Pedneka.et.al [7]** In this paper, the author has proposed Viterbi decoder and convolutional encoder with constraint length 3 and code rate of  $\frac{1}{2}$ . The authors have realized the proposed design using Matlab. For Viterbi algorithm, maximum likelihood algorithm has been used by the authors. The basic implementation and design of convolutional encoder and Viterbi decoder with the help of Viterbi algorithm has been explained by the authors. This paper explains step by step working of three basic modules (BMU, PMU and SMU) of Viterbi decoder. It can be seen that without coding, the value of  $E_b/N_o$  needed to achieve a BER of  $10^{-3}$  is around 12.2Db. This error rate can be achieved with coding at around 9.9db using hard decision Viterbi decoder. So we can express it by saying that the coding gain at a BER of  $E_b/N_o$  is 2.2db. This paper helps beginners to understand working of Viterbi algorithm those who wants to work on Viterbi decoder.

**Bineeta Soreng. et.al [8]** In this paper, the authors has described the convolutional encoder and viterbi decoder maximum likelihood algorithm. The main of the authors are to achieve the encoding and decoding rates as WiMAX standards. This paper uses the constraint length of 7 and generator polynomial (171, 133) and implemented on EP4SGX70HF35C2 device in Alters DE board. VHDL coding is being used. The proposed design has been implemented and synthesized using Altera Quartus II vl 1.0 and is simulated using ModelSim Altera Starter Edition 6.6d. We have implemented area optimized VLSI architecture of Convolution Encoder and Viterbi Decoder. The proposed architecture is implemented in EP4SGX70HF35C2 device of Stratix IV family and realized using ModelSim 6.6d. It shows area improvement in terms of hardware utilized. The silicon validation is done by programming the FPGA with VHDL code of the proposed structure implementation.

**Yin Sweet Wong. et.al [9]** The authors in their paper worked on the realization of AVD (adaptive Viterbi Decoder) with constraint length of 3 and code rate of half using the technology of Field programmable gate array. They focused on the 4-state, hard decision Adaptive Viterbi Decoder which can results in the decoding through different trace back length. This has been implemented and performance has been analyzed using ISE 9.2 and simulators called MATLAB. The authors targets on the improvement of over 20%. A basic convolutional encoder and an adaptive Viterbi decoder have presented in this paper. The adaptive ability of the Viterbi decoder is according to the threshold level for BER of  $10^{-3}$ . The two different models of Viterbi decoder are according to the TL of the trellis paths. Both models are synthesized and programmed to the FPGA board. The purposed architecture enforces a simple but highly efficient Viterbi arithmetic unit which helps the device utilization or silicon area. Custom implementations in FPGA, allows real time processing with lower delay (BER of  $10^{-3}$ ) and non realtime processing with higher delay (BER of  $10^{-6}$ ), providing a good trade-off between performance and area.

**Mahe Jabeen.et.al [10]** Authors presents an error correction codes to limit the effects of noise in digital communication. Authors employed the convolutional codes to implement FEC but complexity increases exponentially with constraint length. They realized the encoder of 3 constraint length and half code rate using Verilog HDL. They simulate it using Modelsim Altera 10d and ISE 10.1. This architecture has

comparatively simpler code and flexible configuration when compared to other architectures and saves silicon area through efficient device utilization which makes it favorable for present day FPGA's. Although this architecture has limitations because of the increasing number of computations in decoding performed at each stage which makes it impractical for convolution codes with large constraint length, it provides a good tradeoff between performance and area.

**Yan Sun.et.al [11]** The authors in this paper represents modified scheme for the Viterbi decoder and convolutional encoder using standards based on IEEE 802.11a. Here the convolutional encoder uses  $k$  of 7 and code rate of  $\frac{1}{2}$  and decoding depth of 36 symbols. Designed Viterbi decoder uses full-parallel structure to improve Add-compare-select computational speed, optimal data storage mechanism has been adopted to avoid overflow and three distributed RAM blocks are employed to complete cyclic trace-back. The author's presents the design having chip logic elements with 10% decrease, reduction of power consumption with 5% and increase in the encoder decoder working performance over hardware implementation. In this paper, a (2, 1, 7) convolutional encoder is pro-posed for FPGA design and implementation, and then an adaptive, low-power, parallel Viterbi decoder with a constraint length of 7 and a code rate of  $\frac{1}{2}$  is presented. The designs are functionally verified in Xilinx ISE 7.1 environments. The synthesis results show that the FPGA implementation can run with frequency up to 80 MHz. Compared to the solutions with generic decoder, the proposed scheme uses significantly less logic resources on the FPGA. The design takes about 2.94% of the total chip logic elements. The maximum operating frequency is 80 MHz that is found adequate to our applications.

**J.Tulasi.et.al [12]** Authors present their concern on design and implementation on Viterbi decoder and convolutional encoder using FPGA technology. Deep space communications and wireless digital communication use convolutional encoder which is an essential part in term of communication. The main aim of this paper is to realize a Viterbi decoder of constraint length of 9 and half code rate using Xilinx 12.4i tools. The Convolutional encoder for the constraint length of  $K=9$  and code rate of  $r=1/2$  has been developed and the synthesis is carried out. It has been simulated and the simulation result was shown in the paper. The Viterbi decoder has been developed

using and the synthesis is carried out. It (decoder) has been simulated and the simulation result was shown in the paper.

Typical input and output are as indicated below.

Input bits X = [010111001010001]

Output bits Y = [00 11 10 00 01 10 01 11 11 10 00 10 11 00 11]

Encoded noise = [00 11 11 00 01 10 01 11 11 10 10 10 11 00 11]

**Hiral Pujara.et.al [13]** The authors in this paper aimed to describe the comparative study / analysis between various FPGA devices for the implementation of optimized Viterbi decoder. Simulation and synthesized result was the base of comparison. Trace back architecture was used to design optimized Viterbi decoder. Viterbi decoder with half rate and encoder with k of 3 were designed using VHDL simulator and synthesized using XST tool. In this Paper Resource optimized Viterbi Decoder has been proposed. The proposed Viterbi Decoder has been designed with VHDL using traceback method. The designed Viterbi Decoder has been simulated using Xilinx ISE simulator and synthesized with XST. The simulated and synthesized results show that proposed design can work at an estimated frequency of 33.124 MHz by using considerable less resources of target FPGA device SPARTAN 3A. This Paper also contains comparative analysis between various FPGA devices for the same Design. The result shows that proposed design can work at Max. Frequency 113.104 MHz for targeted FPGA Device VIRTEX 5 among all FPGA Devices. So, VIRTEX 5 FPGA Device can give maximum frequency for proposed Design among all FPGA Devices.

**Sherif Welsen Shaker.et.al [14]** The authors laid emphasis on the FPGA for implementing many complex and sophisticated signal processing works and tasks using SDR (software defined radios). Robust digital communication system uses the convolutional codes and in wireless communication Viterbi decoding algorithm was used to decode the convolutional encoder. But these are very complex and consume large amount of power. Using a VHDL code for FPGA implementation a low power-reconfigurable Viterbi decoder is implemented for WiMAX receiver. Features like higher flexibility, re-configurability and shorter time to market give the FPGA new opportunities for the effective insertion in SDR conditioning chain. In this paper, a design of low power, configurable Viterbi decoder for WiMAX receiver has been proposed. The design benefits the concept of trace-back approach with clock gating

for power reduction. The power report for the proposed design have been driven using Xilinx XPower Analyzer tool, and it showed the reduction in the dynamic power dissipation which is a good indication for power reduction when implementing the proposed design on ASIC. The design took about 7% of the total chip logic elements. The maximum operating frequency is 47.4MHZ, which is found adequate to the proposed design.

**Dayanand B M.et.al [15]** The authors implementing the Viterbi decoder and convolutional encoder by considering the issue of reliability and data transmission efficiency. The main aim was to realize convolutional encoder with  $k$  of 3 and decoder with code rate of half. The maximum likelihood Viterbi decoder algorithm was used to design the proposed design. The whole design was realized using MATLAB. This paper explains step by step working of three basic modules (BMU, PMU and SMU) of Viterbi decoder. It can be seen that without coding, the value of  $E_b/N_o$  needed to achieve a BER of  $10^{-3}$  is around 12.2Db. This error rate can be achieved with coding at around 9.9db using hard decision Viterbi decoder. So we can express it by saying that the coding gain at a BER of  $E_b/N_o$  is 2.2db. This paper helps beginners to understand working of Viterbi algorithm those who wants to work on Viterbi decoder.

**Erik Normark.et.al [16]** The authors in this paper represents top down design, simulation and design using hardware description language, of complete  $\pi/4$  DQPSK system by using VHDL-AMS. Here two implementations were used with and without Viterbi encoder/decoder. The proposed design was simulated and BER was evaluated in the presence of noise. In this paper, we described a methodology for top down modeling and simulation of complete RF/mixed signal system using VHDL-AMs. As a demonstration example, we considered a  $\pi / D4Q$  PSK system and developed a library of behavioral level blocks for it. We simulated our system in a noisy environment and using Mentor Graphics Advance-MS and Agilent ADS. BER analysis showed that, although some minor differences existed between the VHDL-AMs and Agilent ADS models, the performance of the system very closely matched that of the theoretical model. Following top-down design methodologies, the data source and receiver were made more complex by implementing a simple Viterbi

encoder and decoder in VHDL, resulting in an approximate 1.4dB improvement over theoretical.

**C.Arun.et.al [17]** In this paper author uses a novel approach to design a high output with reduced BER probability Viterbi decoder and implemented it. Low BER (bit error rate) has been achieved by increasing free distance without increasing complexity. This increase in free distance can be achieved by proposed non-polynomial convolutional code method. A novel Viterbi algorithm was designed and implemented using non-polynomial approach at Spartan-III FPGA for achieving low bit error rate and high throughput rate. We have chosen the larger  $d_{free}$  to achieve the best performance of the system. We have made an attempt to study the effect of  $d_{free}$  on constraint length and code rate. The proposed method gives the maximum throughput of 423.93 Mbps at the encoder section and 84.93Mbps at the decoder section of the architecture. The number of FF used for the above encoder and decoder are 7 and 518 for constraint length of 3 and code rate  $\frac{1}{4}$ , respectively. The probability of error for the constraint length of 3 and code rate of  $\frac{1}{4}$  is  $8.12 \times 10^{-10}$ . We have achieved the overall performance of the Viterbi decoder architecture for the proposed method is 75% than the existing value of 25%.

Table 2.2 Review of Previous Architectures [15]

ARCHITECTURE	RESULT
<i>Convolutional encoder and Viterbi decoder in DSP platform</i>	Flexible platform but slow in speed.
<i>Convolutional encoder and Viterbi decoder in micro-controller platform</i>	Slow speed.
<i>FPGA based Convolutional encoder and Viterbi decoder.</i>	Complexity of Viterbi algorithm increases.

## **2.1 Conclusion:-**

From the above survey on the convolutional encoder and Viterbi decoder the following points may be concluded:-

- Convolutional encoder and Viterbi decoder are designed by using known input bits.

- Result of which is that the parameters like efficiency, speed, BER are concluded over that specific known bits.
- Convolutional encoder and Viterbi decoder are implemented by using deterministic noise formula.
  - This leads to the result that deterministic noise formula cannot be said as a noise.
- Implementation of convolutional encoder and Viterbi decoder without using modulator.
  - By not using any modulation technique, design cannot be implemented for real time applications.

By considering the above said gaps in literature review, a design is proposed which has following new work.

- ✓ Implementation of convolutional encoder and Viterbi decoder has been done by giving random bits to the input of encoder.
- ✓ Random noise channel AWGN has been generated by the use of Box Muller Method.
- ✓ QPSK modulation technique has been used to modulate the encoded output bits over the AWGN channel.

### 3. CONVOLUTIONAL ENCODER AND VITERBI DECODER

Communication means transfer of information from the transmitter end to the receiver end through any medium. The medium may be air, wired. Due to the presence of noise, at the receiver end information will get noisy and actual information will not be received. So, at the receiver end there will be some errors. These errors can be detected and sometimes it's also possible that these errors can be corrected using the coding techniques.

#### 3.1 Coding

The error control methods are generally of two types:-

- Error detection with retransmission
- Forward error correction

In error detection with retransmission method, when the receiver end detects the error acknowledgement is sent to the transmitter to send the data again as the previous was corrupted. Whereas in forward error correction method, proper coding techniques are used to detect and correct the errors.

Coding technology uses a proper techniques of adding some extra bits to the transmitted message bits and these redundant bits are used to detect errors at the receiver side. But addition of redundant bits results in decrease in bit rate at the transmitter and its power. But also have advantage that probability of error is less.

Extra bits are added with the original message signal by the channel encoder. Whereas channel decoder finds these added extra bits and with the help of these bits detect errors and correct the corrupted signal. If 'k' are the bits in one block of original message and during transmission extra bits are added to this block and total encoded message bits become 'n'. This becomes a block code ( n, k ).

Thus we can say that,

Extra bits added by the encoder to the block =  $n - k$

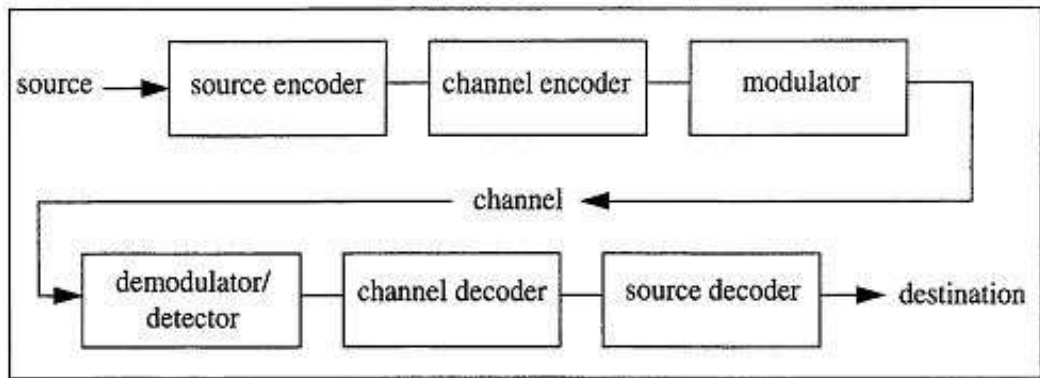


Figure 3.1 Communication System with channel encoding / decoding operations

### 3.1.1 Block Coding

In this technology, fixed numbers of message symbols are mapped by a fixed number of code symbols. During encoding, each block is treated separately and as a memory less device.

### 3.1.2 Convolutional Coding

In this technology, a sequence of message symbols are accepted which resulted in sequence of code symbols. Its output doesn't depend on the current input but also on the previous input bit sequence.

## 3.2 Convolutional Encoder

The basics features of convolutional encoder are

- A finite state machine
- Have fixed number of shift registers
- At every clock, bit enters to the next shift register
- Number of output bits depends on modulo-2 adders used.

### 3.2.1 Encoder parameters

There are three parameters used in convolutional encoder (n, k, m)

where,[2]

m = no. of shift registers used

k = no. of bits input to the encoder

n = no. of output bits from encoder

Bandwidth efficiency of the code is measured by the quantity  $k/n$ . This is also known as code rate.

Table 3.1 Parameter Range for Convolutional Encoder

<u>Parameter</u>	<u>Range</u>
K	1 to 8
N	1 to 8
M	2 to 10
Code rate	$1/8$ to $7/8$
Code rate for deep space application	$1/100$

There is another representation for convolutional codes. Its  $(n, k, L)$ , where, L is said to be the constraint length of the code. L is also mathematically equal to the  $m + 1$ .

In this thesis convolutional encoder will be discussed as  $(n, k, L)$ .

### 3.2.2 Encoder structure

The encoder structure is usually consists of generator polynomial (g). The number of output bits in the encoder depends on the Modulo-2-adders used. But the what will be the result for individuals output bits depend on the selection of the shift register to be modulo-2 operated and that's known as the generator polynomial for individual output bit. Now how to select the shift registers, that found by the hit and trial method and side-by side checking the error through simulation on computer. In this thesis, the polynomial used is just finding out by hit and trial method and checking the performance characteristics.

**i. Convolutional coder  $k = 2, r = \frac{1}{2}$**

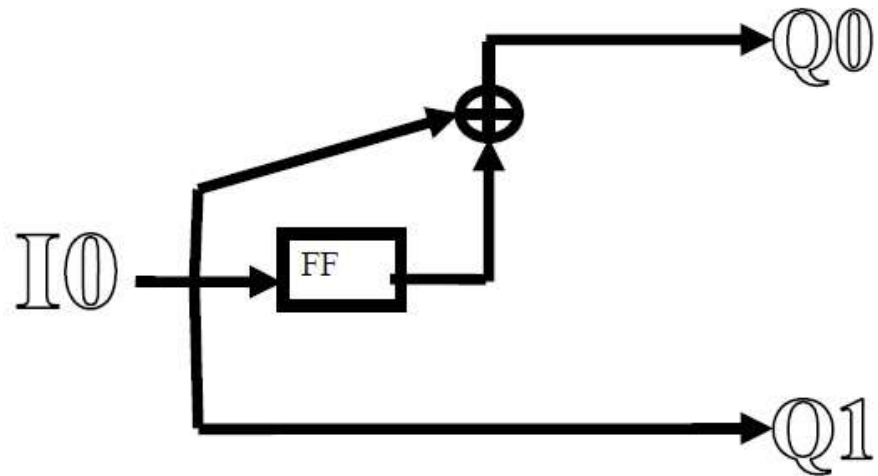


Figure 3.2 Convolutional coder  $k = 2, r = \frac{1}{2}$

$$Q0 \Rightarrow G^{(0)}D = 1 + D$$

$$Q1 \Rightarrow G^{(1)}D = 1$$

**ii. Convolutional coder  $k = 3, r = \frac{1}{2}$**

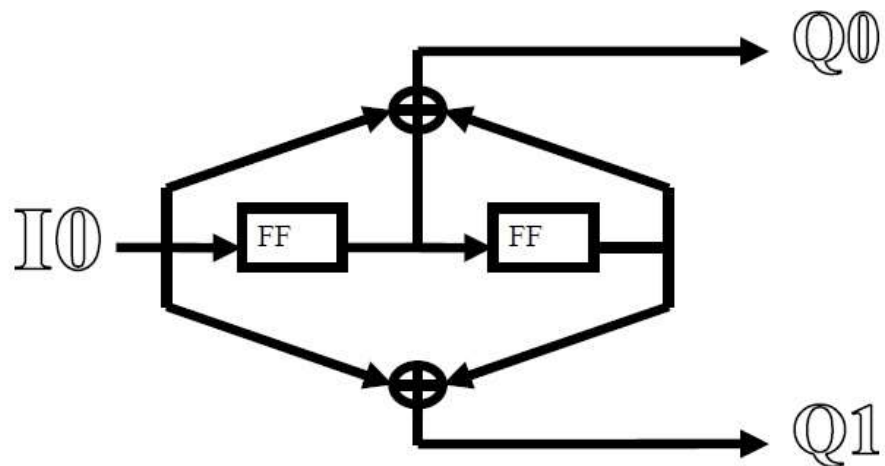


Figure 3.3 Convolutional coder  $k = 3, r = \frac{1}{2}$

$$Q0 \Rightarrow G^{(0)}D = 1 + D + D^2$$

$$Q1 \Rightarrow G^{(1)}D = 1 + D^2$$

iii. Convolutional coder  $k = 5, r = \frac{1}{2}$

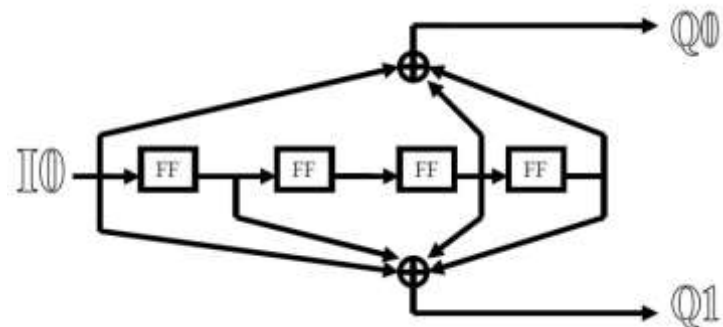


Figure 3.4 Convolutional coder  $k = 5, r = \frac{1}{2}$

$$Q0 \Rightarrow G^{(0)}D = 1 + D^3 + D^4$$

$$Q1 \Rightarrow G^{(1)}D = 1 + D + D^3 + D^4$$

3.2.3 Encoder states

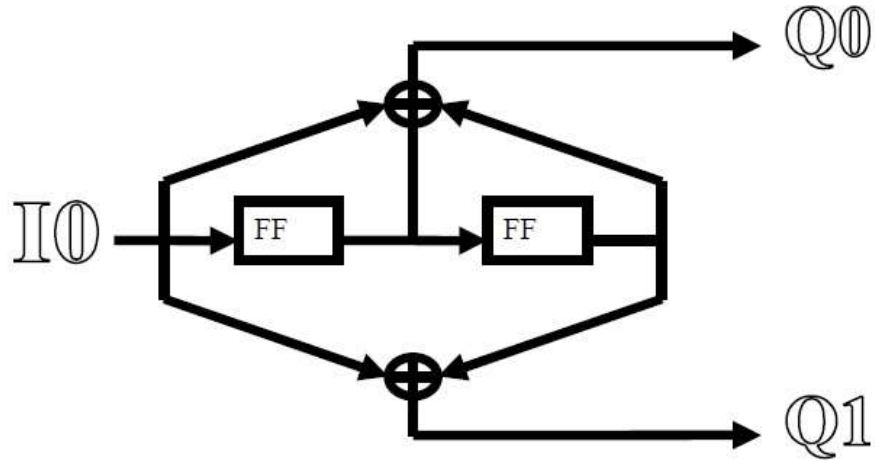


Figure 3.5 Convolutional coder  $k = 3, r = \frac{1}{2}$

In this one input, 2 flip-flop and two output then convolutional encoder can be represented as (2, 1, 3). Constraint length of the given convolutional encoder is 3. Incoming bit is kept by the first memory register and the previous input bits according to the clock are kept by other shift register. This determines the state of the code. There are 2 output bits so there will be four possible states that may be 00,01,10,11.

**3.3 Special Case Convolutional Encoder**

3.3.1 Punctured Codes

- Higher rate can be achieved using puncturing coding

- Some transmitted bits are permanently deleted by encoder.[4]
- There are 3 rates
  - For  $\frac{1}{2}$  no bit will be deleted.
  - For  $\frac{2}{3}$ , every fourth incoming bit of 4 bit is to be deleted for achieving this much rate.

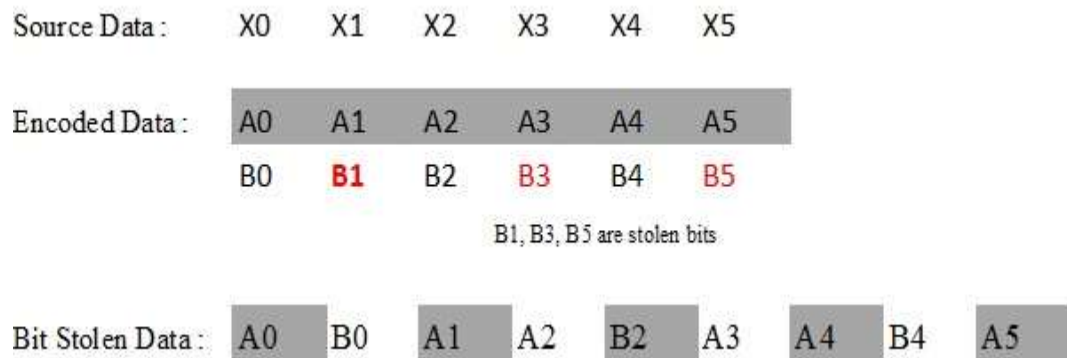


Figure 3.6 2/3 Puncture Coding

- For  $\frac{3}{4}$ , every third and fifth incoming bit of 6 bit is to be deleted for achieving this rate.

### 3.3.2 Dual – K Codes

- Non – binary convolutional coding.
- Easily decodable using Viterbi algorithm.[7]
- Stages are  $2k$  – bit shift register.
  - Example of dual 3 – bit coding
    - Code rate will be  $\frac{3}{4}$
    - On every clock pulse, previous input bits are shifted to the next shift register.
    - 4 - Bit output is produced at every clock pulse.
    - Code rate of other than  $\frac{1}{n}$  can be achieved.

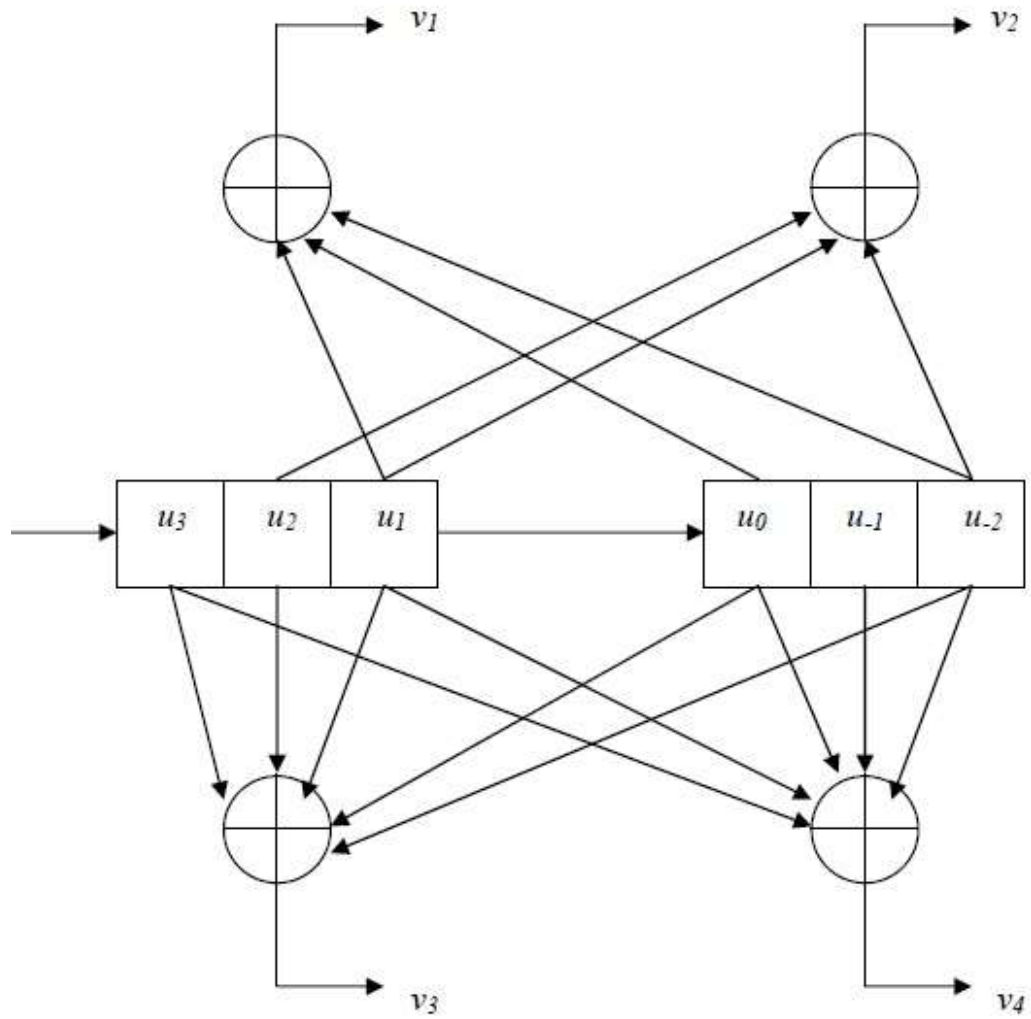


Figure 3.7 A Dual – 3 Convolutional Code

### 3.3.3 Systematic v/s Non - Systematic Codes

- In systematic codes, at the output end, bits received are same as that the input bit sequence.[5]
- Systematic code has less hardware complexity, so more preferable.
- Moreover systematic codes are non-catastrophic.
- Error-protecting properties are same for both systematic and non-systematic coding technologies.

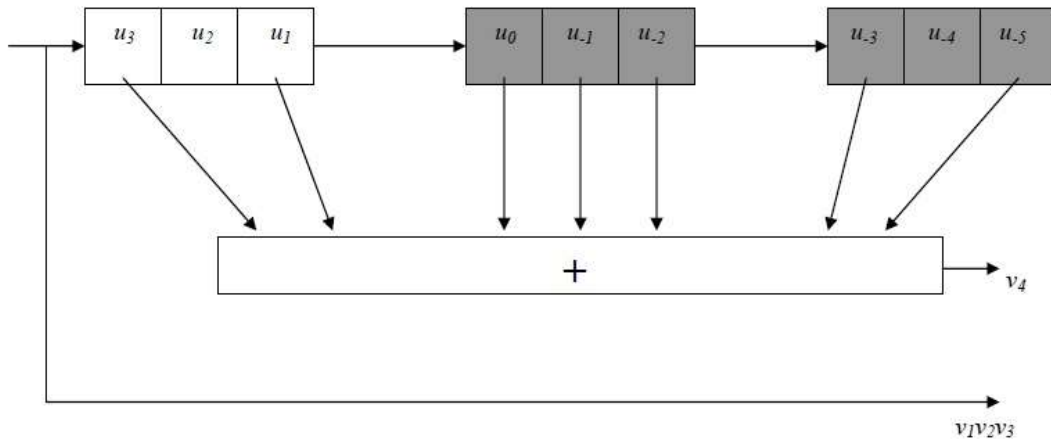


Figure 3.8 (4, 3, 3) Systematic Convolutional Code

As the systematic code name gives a clear indication that output bits will follow the input bits then here also 3 input bits will show its image at 3 output bits but the last 4<sup>th</sup> bit will be the parity bit which is the result of the combination of shift registers. [6]

### 3.4 Encoder Operation

Consider a convolutional encoder with  $k = 3$  and  $r = \frac{1}{2}$ .

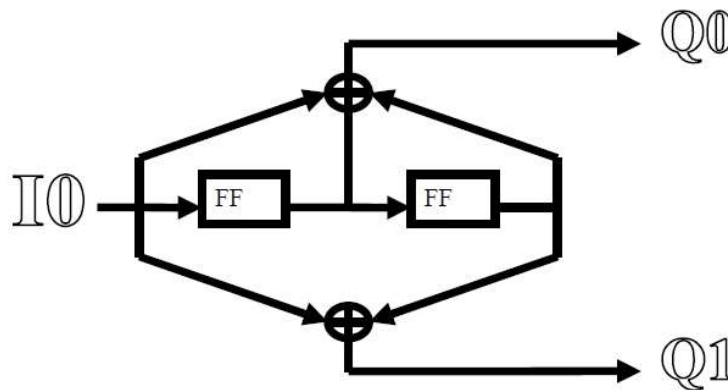


Figure 3.9 Convolutional coder  $k = 3$ ,  $r = \frac{1}{2}$

In this figure,

$I_0 \Rightarrow$  port from where original message signal is to be sent

$Q_0, Q_1 \Rightarrow$  ports from where output bits will be taken out from the encoder.

In the middle there are two boxes which represent the shift or memory registers.

The circle represents the modulo-2 adders. The modulo-2 adders are represented by the means of exclusive-or gate.

A	B	C	O
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Truth Table of Modulo-2 Adder

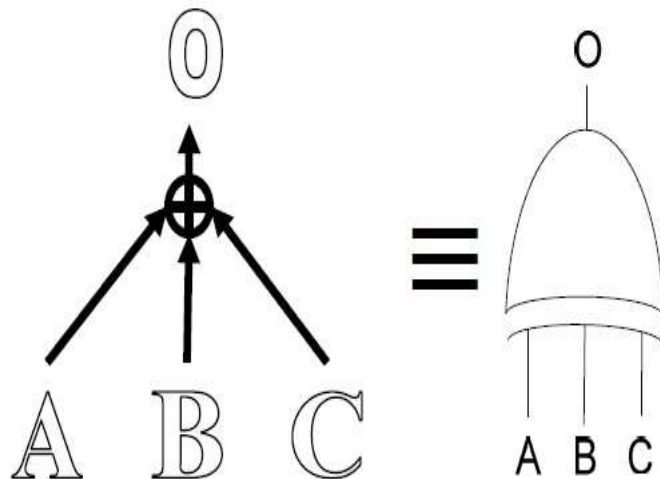


Figure 3.10 Modulo Adder

Operation steps:-

- A. Firstly the encoder is at reset state. That means all the shift registers have initially a bit '0'. Now when a bit enters through the input port I0, this will result in the production of two output bits one each for Q0 and Q1, these are called encoded bits. These output bits generated after the modulo-2 addition of the selected shift(or memory) registers bit with the input bit.[8], [9]
- B. Now "011001110" bit sequence is applied at the input port of the encoder.

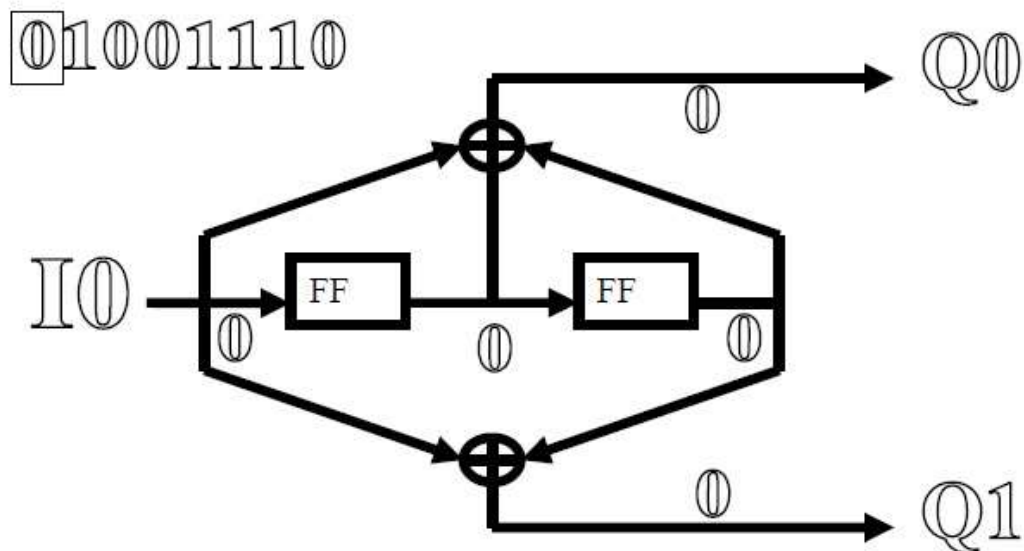


Figure 3.11 Convolutional Encoding Example with 1<sup>st</sup> Bit

Initial state due to reset condition of the shift registers is "00". Now when first bit '0' is input to the encoder then output bits will result in "00". So "00" will be the next state.

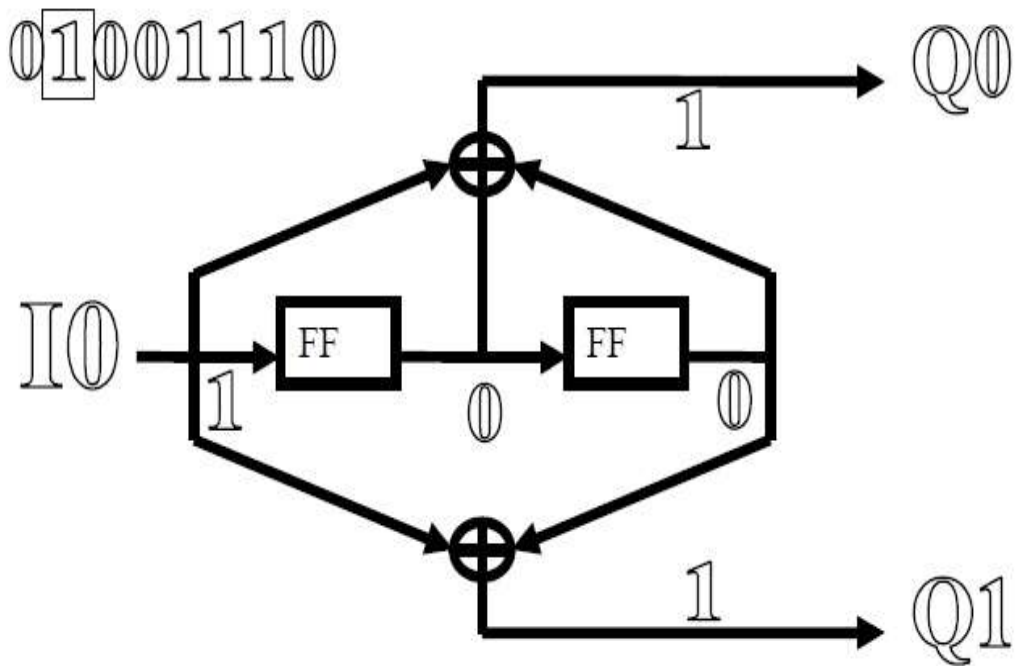


Figure 3.12 Convolutional Encoding Example with 2<sup>nd</sup> Bit

Present state = "11", Next state = "10" when the 2<sup>nd</sup> bit entered into the encoder.

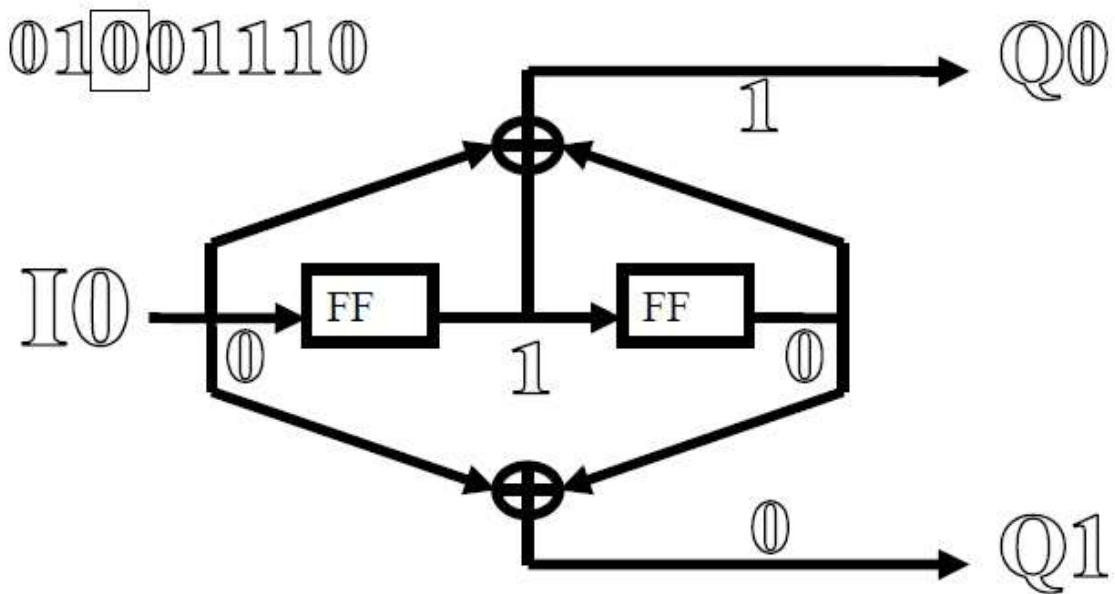


Figure 3.13 Convolutional Encoding Example with 3<sup>rd</sup> Bit

When the 3<sup>rd</sup> input bit entered into the encoder then Present state was "10" and next state will be "01".

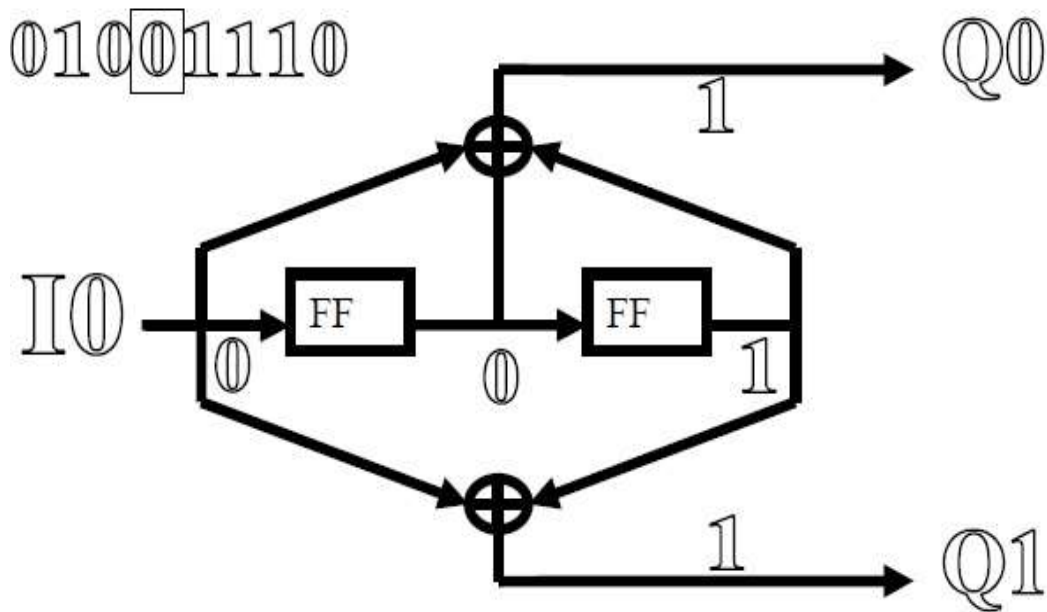


Figure 3.14 Convolutional Encoding Example with 4<sup>th</sup> Bit

As the 4<sup>th</sup> bit entered into the encoder then the present state was “01” and next state will be “00”.

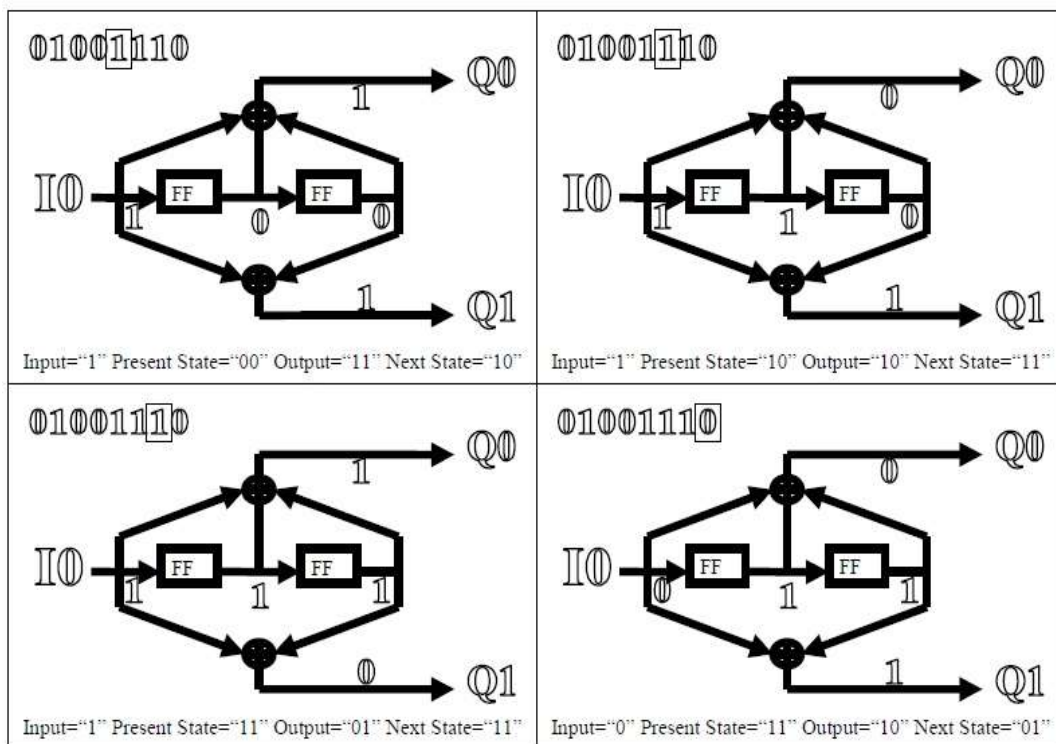


Figure 3.15 Continued example of Convolutional Encoder

Conclusion from the above example [9], [8]:-

- Output is not only dependent on the present bit but also on the previous bits input to the encoder.
- Output pattern got is 0011101111011001.

### **3.5 Convolutional Representation**

To represent in a very simple and easier way, convolutional encoders can be represented in three ways as

- State Diagram Representation
- Tree Diagram Representation
- Trellis Diagram Representation

#### ***3.5.1. State Diagram Representation***

Table 3.2 Convolutional Encoder State Table Representation

<b>INPUT BIT</b>	<b>PRESENT STATE</b>	<b>NEXT STATE</b>	<b>OUTPUT CODEWORD</b>
<b>0</b>	<b>00</b>	<b>00</b>	<b>00</b>
<b>1</b>	<b>00</b>	<b>10</b>	<b>11</b>
<b>0</b>	<b>01</b>	<b>00</b>	<b>11</b>
<b>1</b>	<b>01</b>	<b>10</b>	<b>00</b>
<b>0</b>	<b>10</b>	<b>01</b>	<b>10</b>
<b>1</b>	<b>10</b>	<b>11</b>	<b>01</b>
<b>0</b>	<b>11</b>	<b>01</b>	<b>01</b>
<b>1</b>	<b>11</b>	<b>11</b>	<b>10</b>

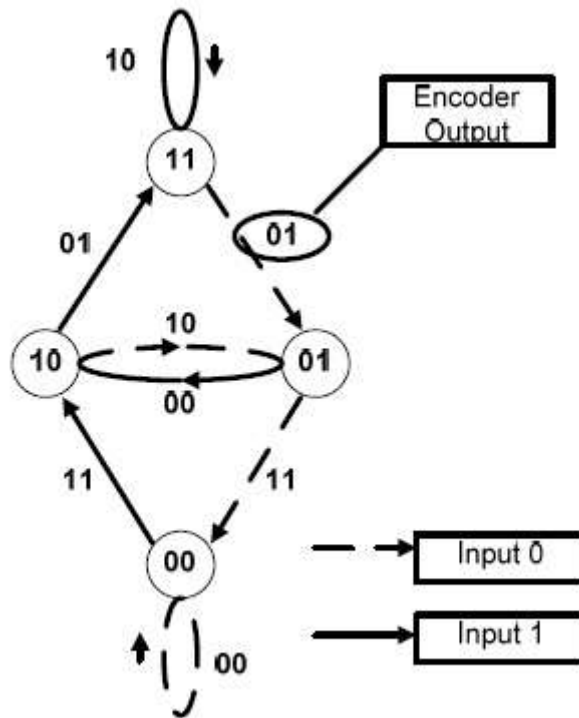


Figure 3.16 Convolutional Encoder State Diagram

In this,

- Bubbles => states
- Inside the bubble => state indicator
- Arrows => state transition according to input
- Value above arrows => output correspond to input during that transition

### 3.5.2. Tree Diagram Representation

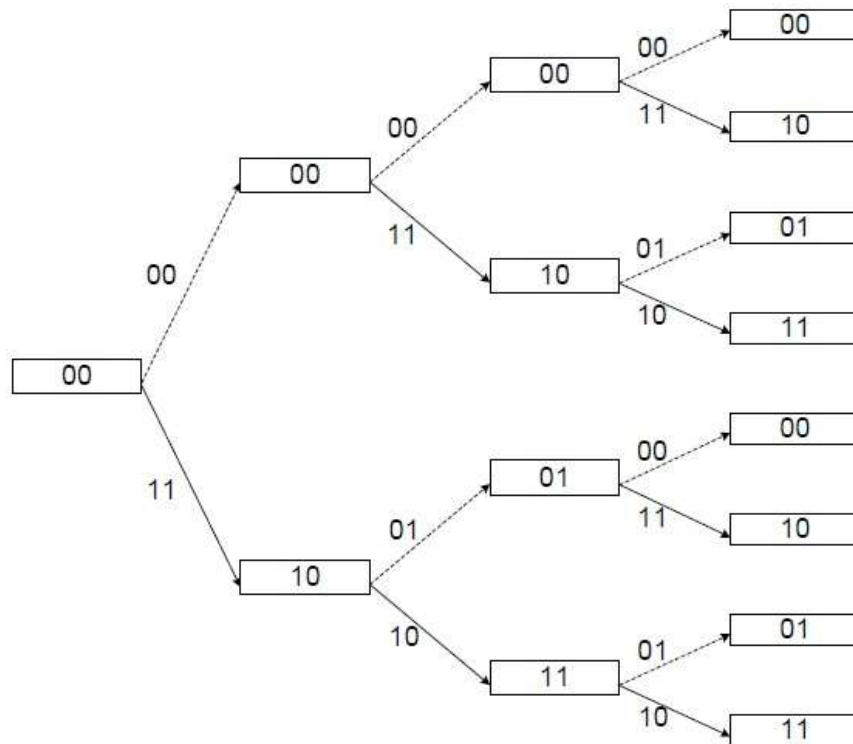


Figure 3.17 Code Tree Representations 1

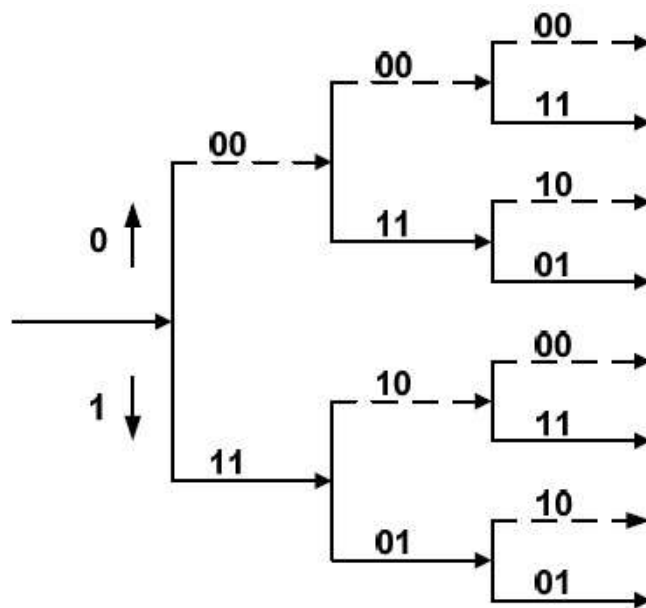


Figure 3.18 Code Tree Representations 2

### 3.5.3. Trellis Diagram Representation

The very simple and ease method of representing convolutional encoder is Trellis Diagram Representation. [22]

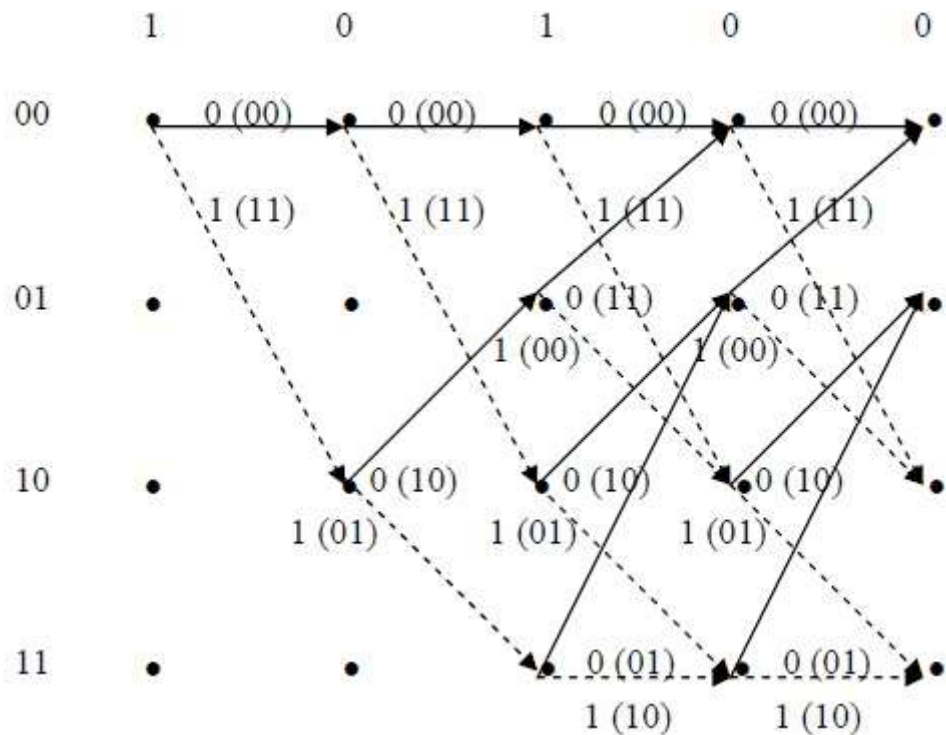


Figure 3.19 Trellis Diagram For (2, 1, 3) Convolutional Encoder

- These are preferred over tree and state diagram both.
- X-axis represents time
- Y- axis all possible states
- With the time, trellis diagram move horizontally i.e. along x-axis.
- At every clock pulse (or transition) there is a arrival of new bit.
- It is unique for every code.
- Initially begins with state “00” and then becomes fully populated.

### 3.6 Decoding

Decoding is the process of retrieving back the original message signal from corrupted or noisy message. There are two types of decoding mechanisms in text book. These are sequential and Viterbi decoding [11], [12]

#### 3.6.1. *Sequential decoding*

- Wozencraft proposed this decoding algorithm
- This was improved by Fano.
- Its main principle is to take one path at an instant.
- That means forward and backward movements are enabled through trellis.
- Variable decoding time is its major drawback.

- With the increase in number of input bits, number of computation increases.

### 3.6.2. *Viterbi decoding*

- Named after Andrew J. Viterbi.
- This Viterbi decoding is also known as maximum likelihood decoding algorithm.
- The drawback of sequential decoding was variable decoding time but in Viterbi decoding this drawback has been removed as here decoding time is fixed.
- Hardware implementation also becomes less complex.
- In sequential only one path at an instant but here whole sequence is examined once.
- For each path, a metric path is chosen by decoder.
- Out of two paths, one path is chosen. That chosen path is called survivors.

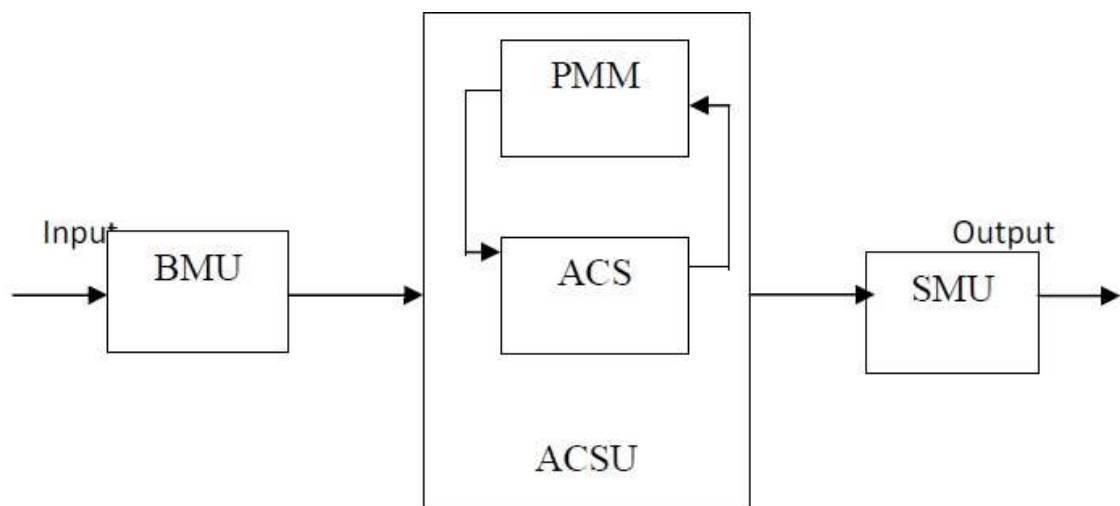


Figure 3.20 Block Diagram of Viterbi Decoder

### 3.7 Hard Decision Viterbi Decoding

- Also known as hard input Viterbi Decoding algorithm.
- Uses Hamming Distance for path calculation.

#### 3.7.1. *Hamming Distance* [13]

The two codes hamming distance is said to be two if their elements differ with exactly two bits.

$$X = 101$$

$$Y = 110$$

Hamming distance = 2

### 3.7.2. Trellis explanation

Consider 3 bit input encoded data [1 0 1]

Suppose the corrupted bits we get are [01 10 00 10 11]

*Step -1 At t = 0,*

- Bit received = “01”
- Decoder starts from “00” initial state.
- There are 2 paths but no one corresponds to the incoming bit sequence.
- Decoder will compute metric for both paths.
- Path metric for both is equal to 1.

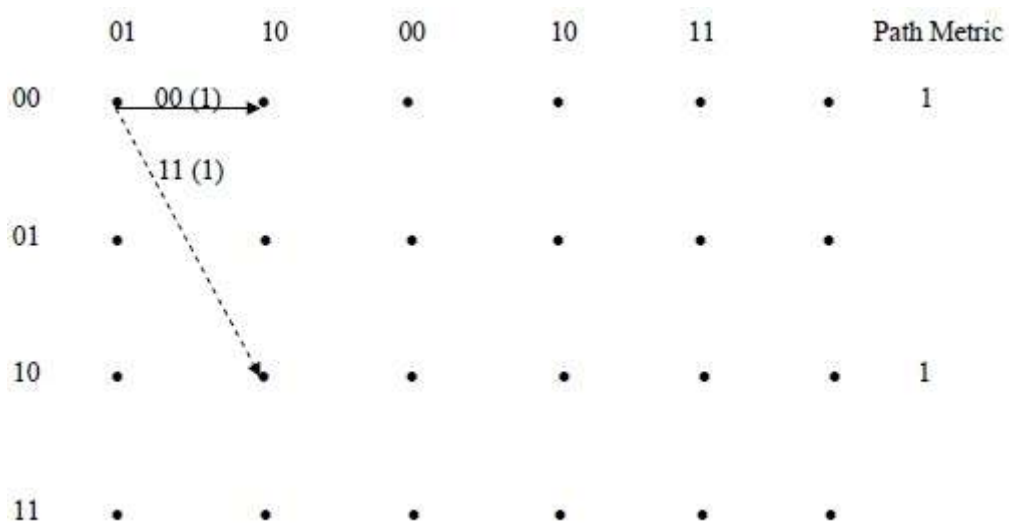


Figure 3.21 Trellis and Path Metrics at time t = 0

*Step -2 At t = 1,*

- Bit received = “10”.
- From last two states, decoder makes computations for all possible 4 states.
- Computation of branch metric is done.
- These computations are added to the previous computations.

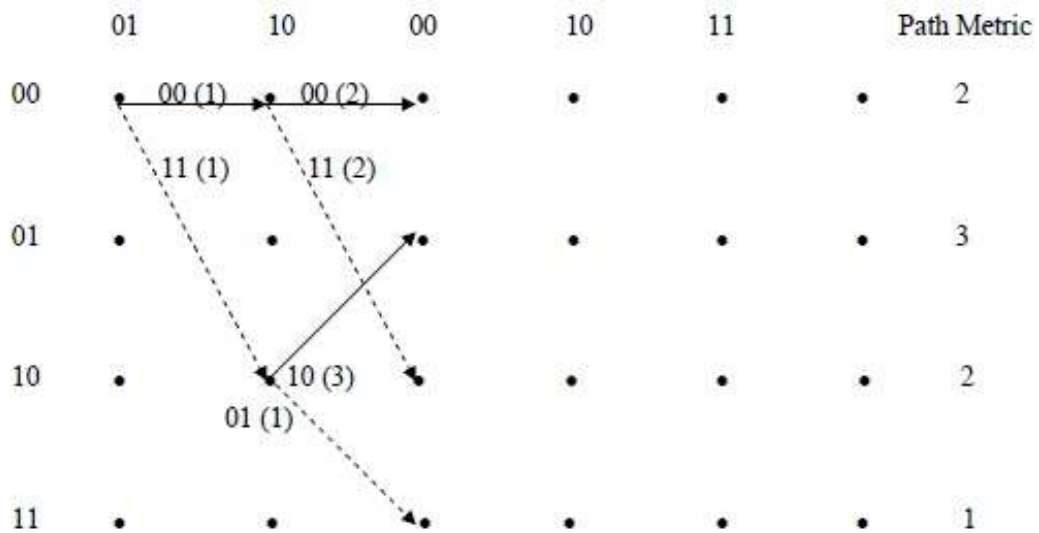


Figure 3.22 Trellis and Path Metrics at time  $t = 1$

**Step 3: at time  $t = 2$ ,**

A. Received bit = “00”

Here use the maximum hamming distance principle.

Discard low metric path at each node.

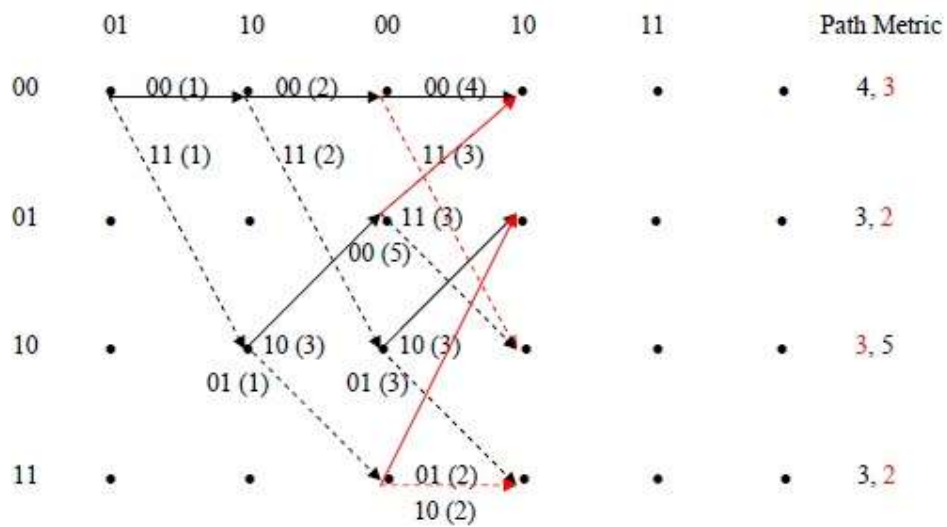


Figure 3.23 Step 3(A) Trellis and Path Metrics at time  $t = 2$

**B.** Discarding will minimize the number of paths.

That will make Viterbi as a least complex algorithm.

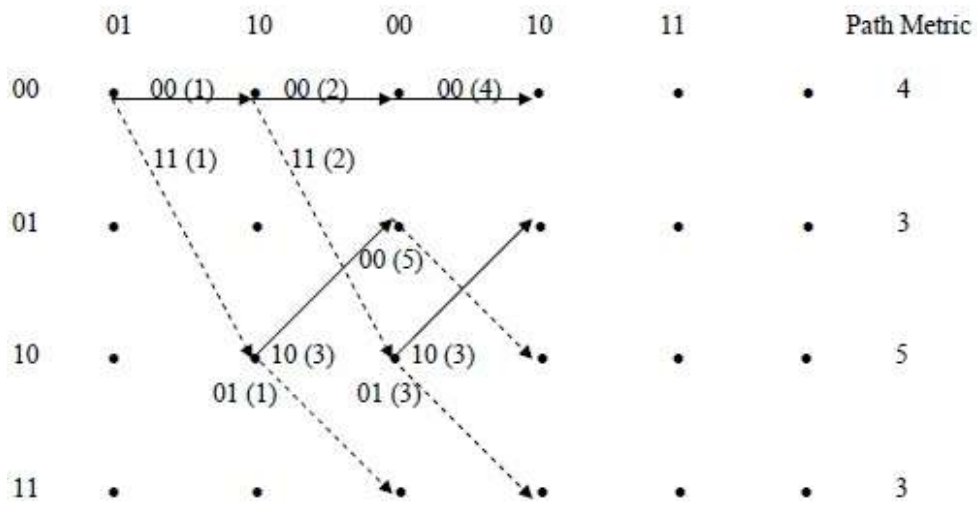


Figure 3.24 Step 3(B) Trellis and Path Metrics at time  $t = 2$

**Step 4: at time  $t = 3$ ,**

**A.** Received bit = “10”

The same procedure goes on.

Computation is done and added to previous branch metric computation.

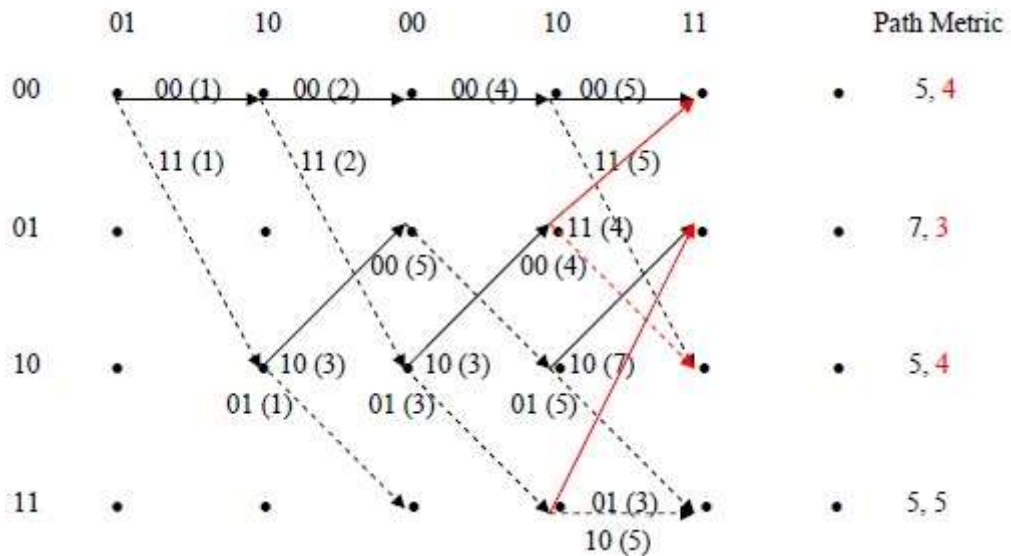


Figure 3.25 Step 4(A) Trellis and Path Metrics at time  $t = 3$

**B.** Discarding will minimize the number of paths.

That will make Viterbi as a least complex algorithm.

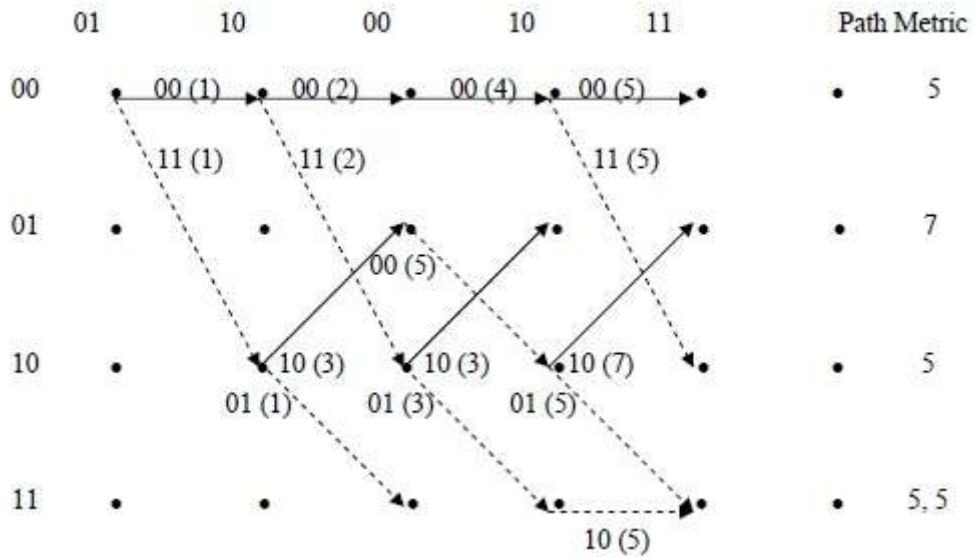


Figure 3.26 Step 4(B) Trellis and Path Metrics at time  $t = 3$

**Step 5: at time  $t = 4$ ,**

**A.** Received bit = “11”

Same procedure is repeated

Trellis is completed here.

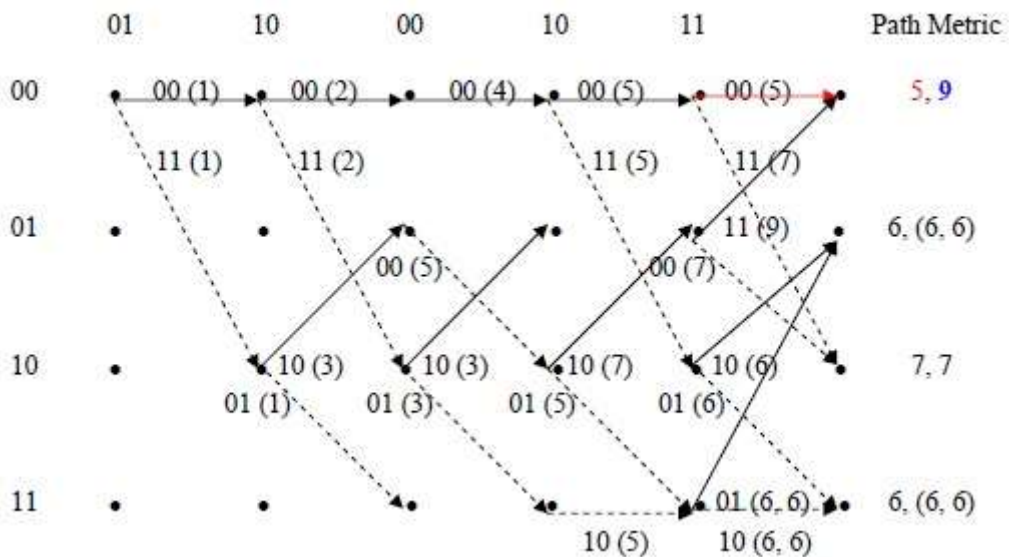


Figure 3.27 Step 5(A) Trellis and Path Metrics at time  $t = 4$

**B.** Discarding of low metric is done and making Viterbi the useful one.

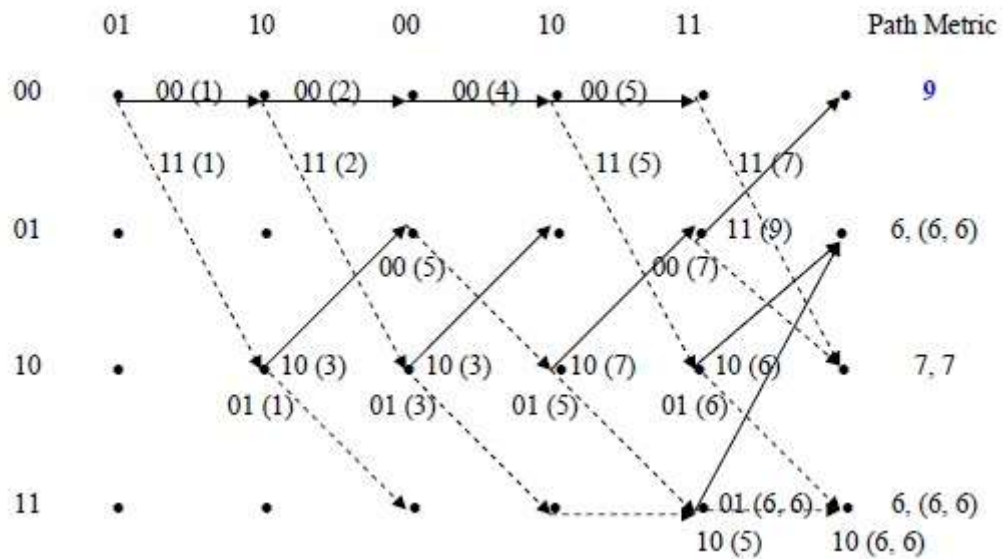


Figure 3.28 Step 5(B) Trellis and Path Metrics at time  $t = 5$

**Final Step:-**

- Highest metric path is kept means maximum hamming distance is kept for decoding. This makes the Viterbi decoder the most useful and efficient one.

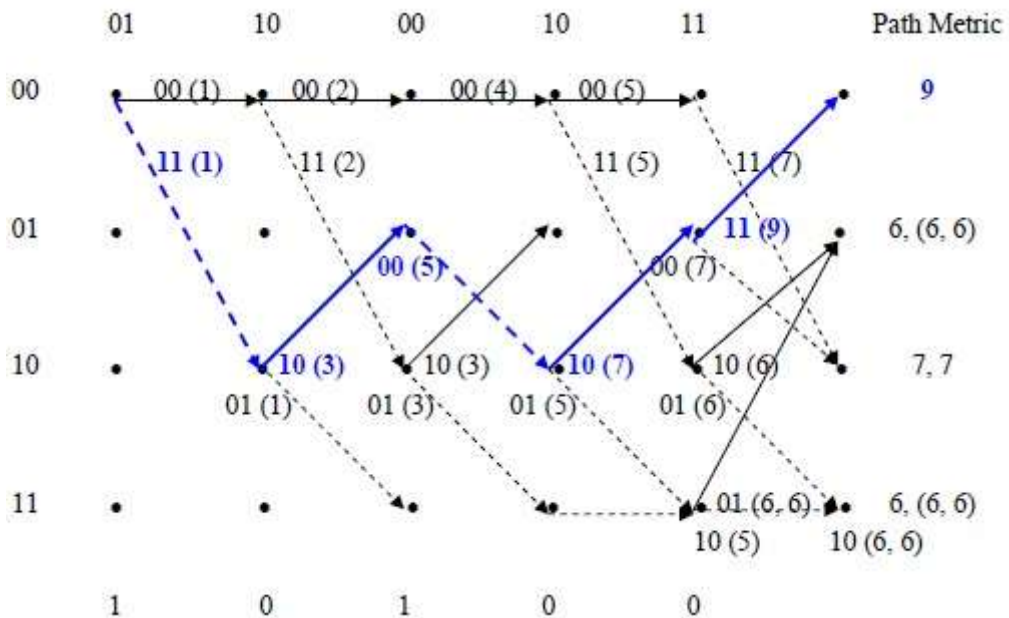


Figure 3.29 Decoded sequence for the noisy channel

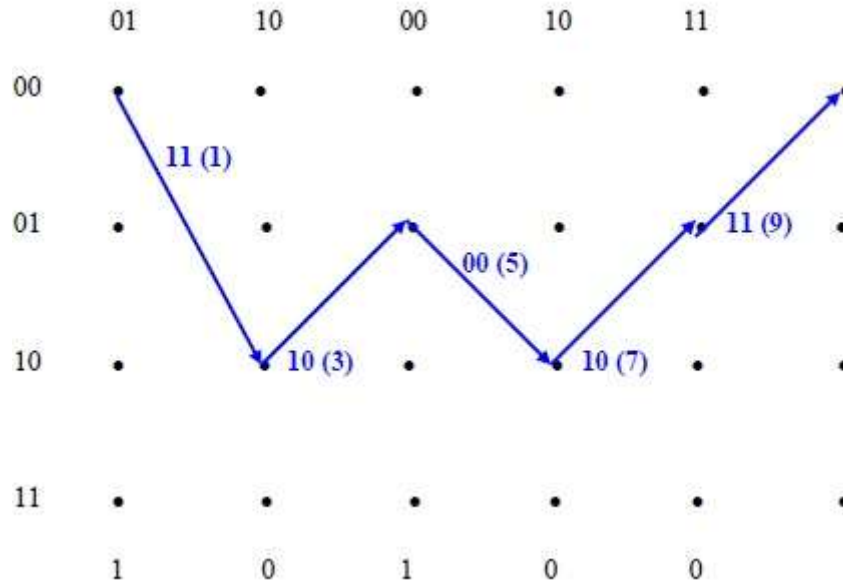


Figure 3.30 Final output

### 3.8 Soft Decision Viterbi Decoding

- Also known as soft input Viterbi Decoding algorithm.
- Uses Euclidean Distance for path calculation.

#### 3.8.1. *Minimum Euclidean Distance*

- It's a means to calculate branch and path metrics in Soft Viterbi Decoding.
- The formula for squared Euclidean Distance is given below:

$$ED(n, i) = \sum [S_k(n) - G_k(n)]^2$$

where,

n = current state

i = input bit

k = encoded bits associated with a given bit

S = received Quantized bits

G = Output bits

The path metrics and branch metrics will be calculated by this formula for Soft Decision Viterbi Decoding.

#### 3.8.2. *Trellis Explanation*

- Same principle will be followed by the soft decision Viterbi decoding as by the hard decision Viterbi decoding.[15], [16]

- The only difference lies in the path metrics and branch metrics calculations.
- Consider 3 bit encoded input data as [1 0 1].
- The output at the encoder is supposed to be [3-4 -43 33 -43 -44]

**Step 1: at t = 0**

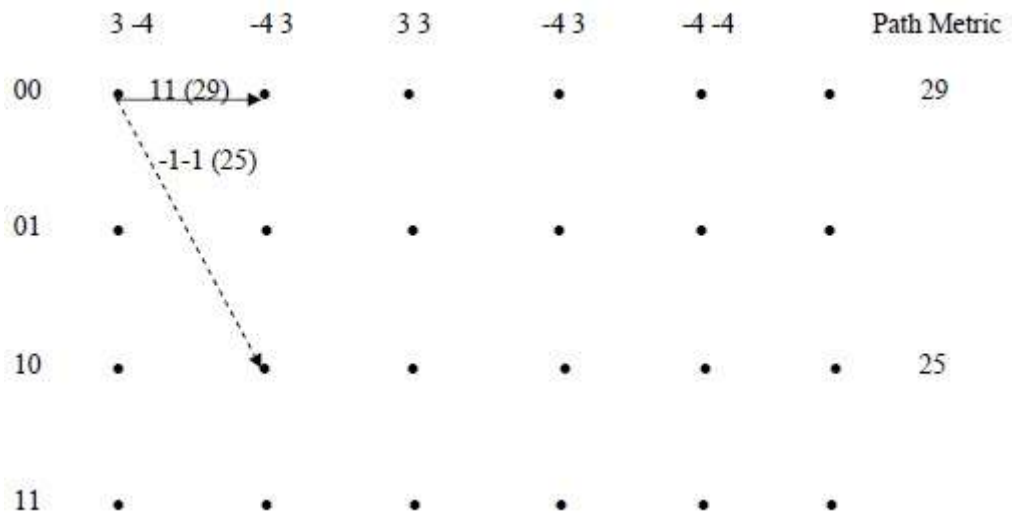


Figure 3.31 Trellis and Path Metrics at time t = 0

**Step 2: at t = 1**

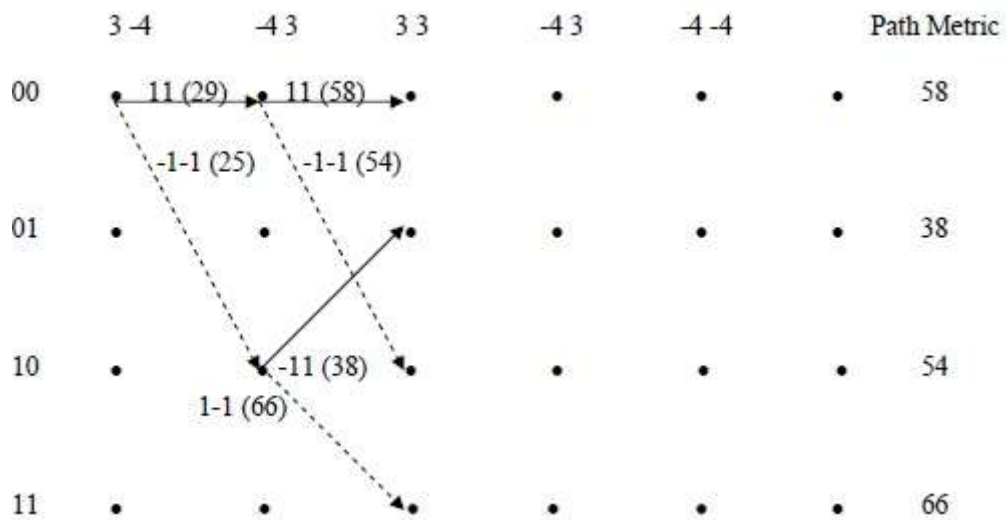


Figure 3.31 Trellis and Path Metrics at time t = 1

**Step 3: at  $t = 2$**

(A)

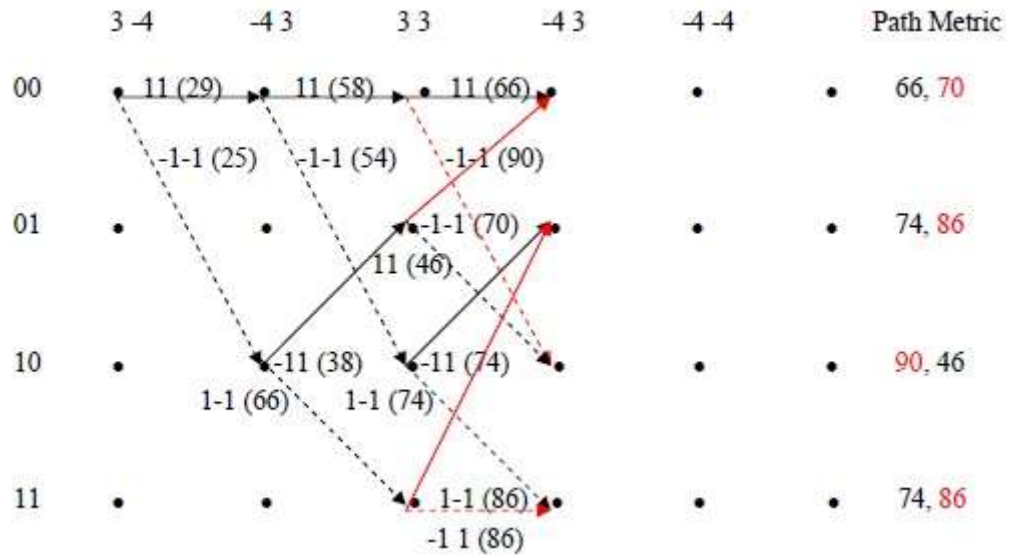


Figure 3.32 Step 2(a) Trellis and Path Metrics at time  $t = 2$

(B)

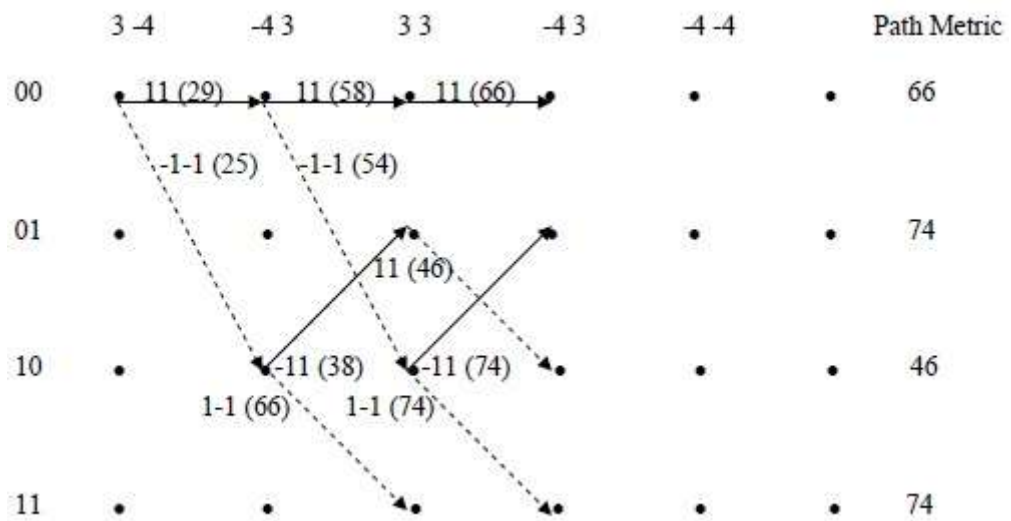


Figure 3.33 Step 2(b) Trellis and Path Metrics at time  $t = 2$

**Step 4: at t = 3**

(A)

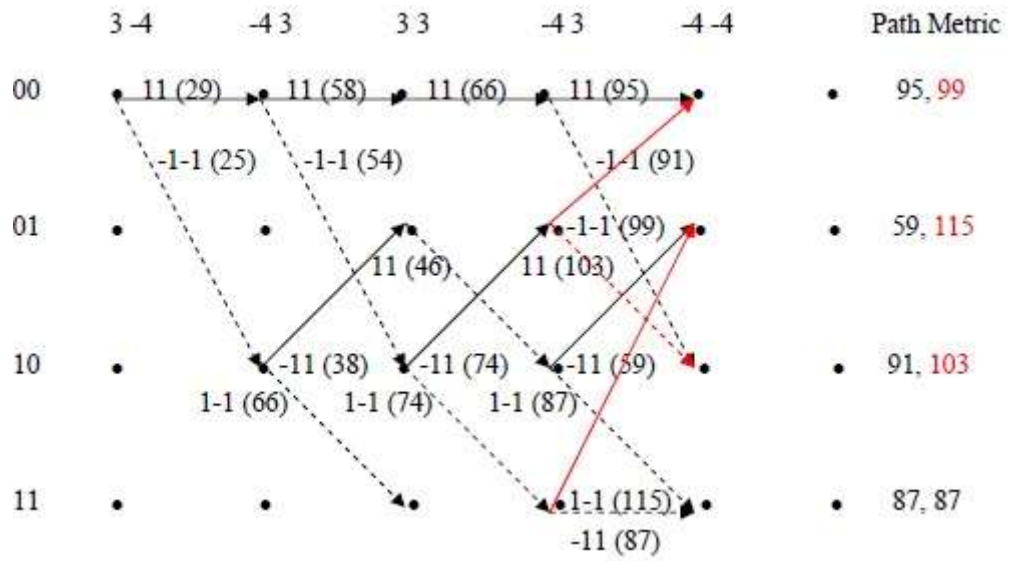


Figure 3.34 Step 3(a) Trellis and Path Metrics at time t = 3

(B)

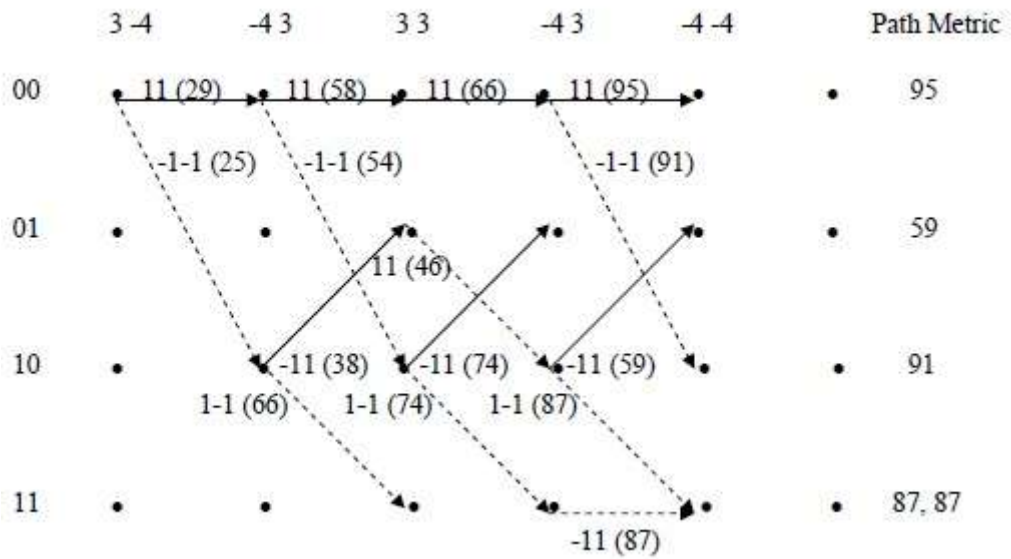


Figure 3.35 Step 3(b) Trellis and Path Metrics at time t = 3

**Step 5: at t = 4**

(A)

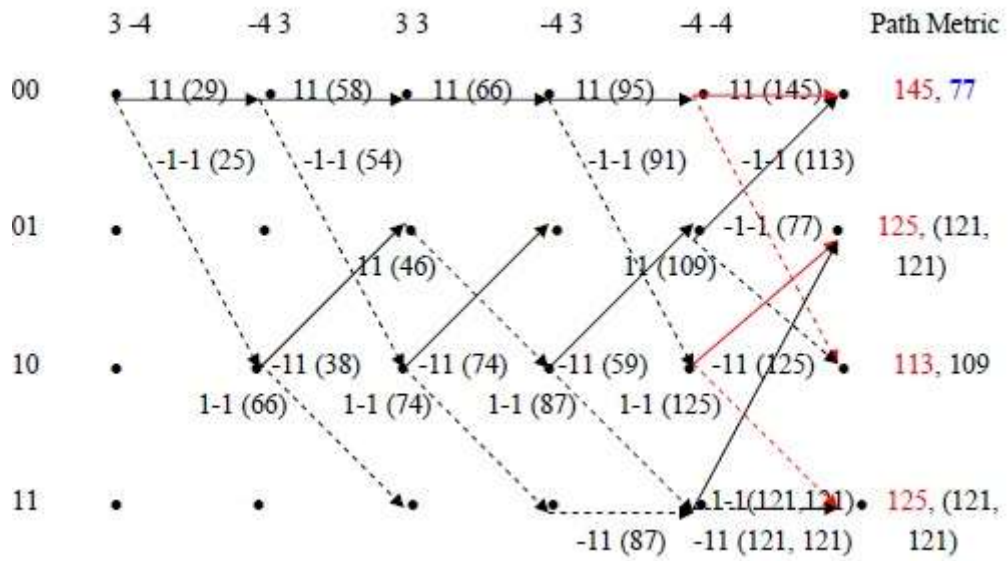


Figure 3.36 Step 5(a) Trellis and Path Metrics at time t = 4

(B)

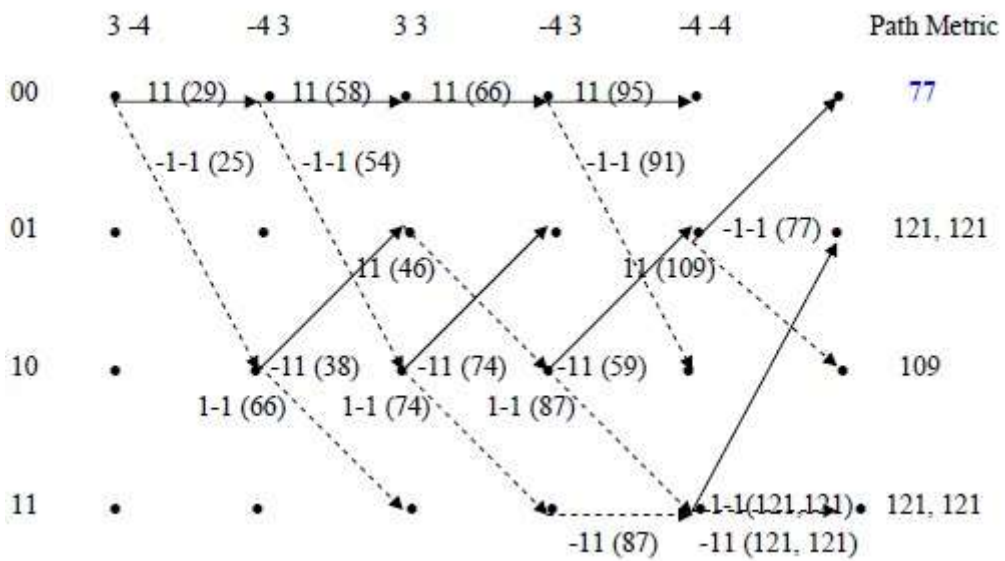


Figure 3.37 Step 5(b) Trellis and Path Metrics at time t = 4

### Step 6: Final Output

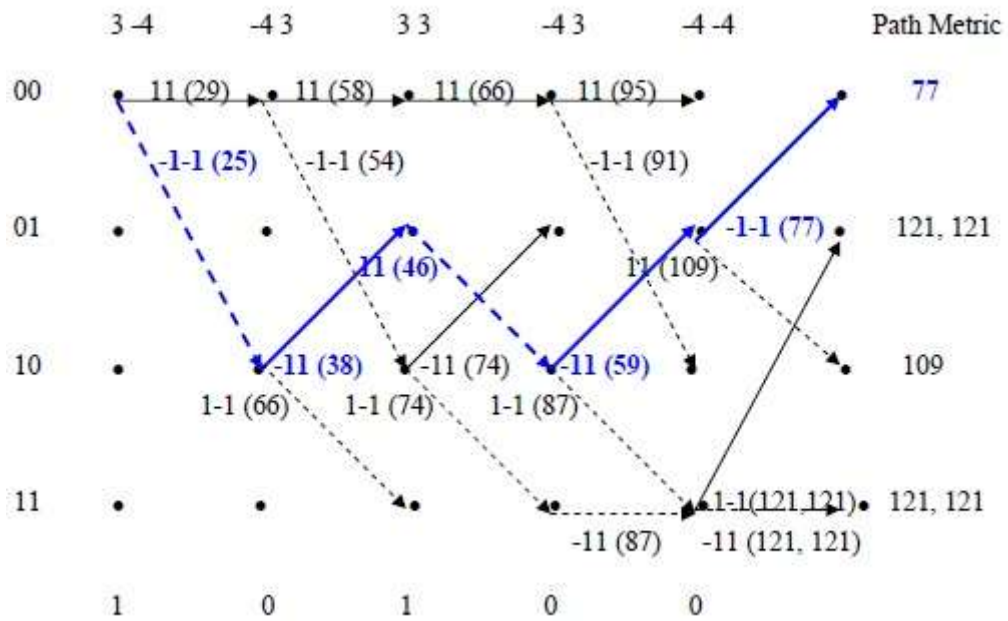
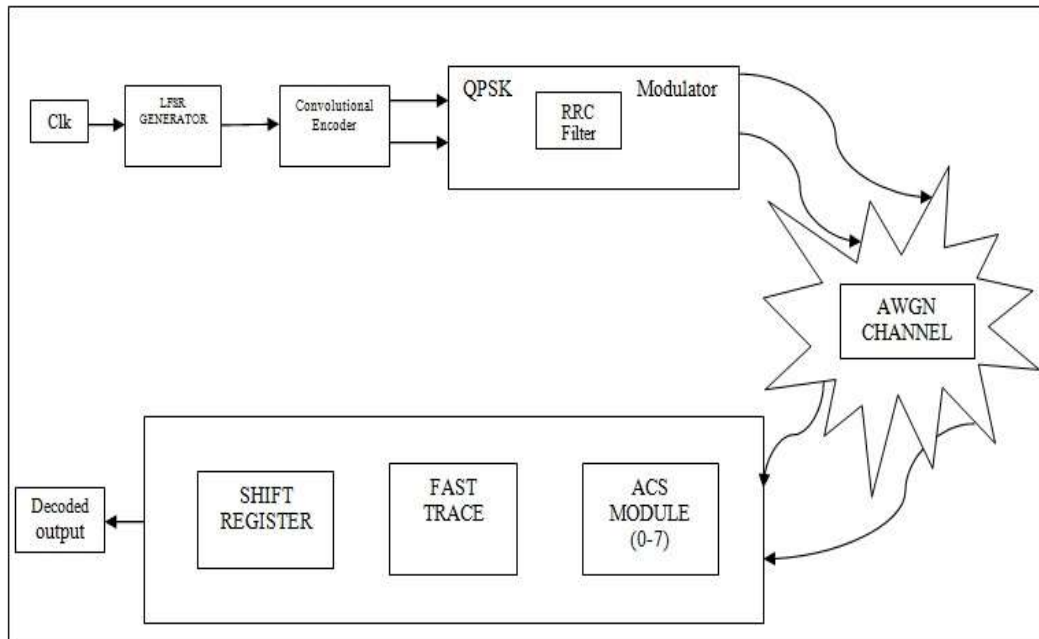


Figure 3.38 Final output

- Path with the minimum distance is traced and called the winner one.
- These figures show how the minimum Euclidean distance works in case of Soft Decision Viterbi Decoder to trace the input bits from the encoder which are transmitted over AWGN channel.

## 4. VHDL AND SDR IMPLEMENTATION OF COMPLETE COMMUNICATION SYSTEM



**PROPOSED DESIGN**

### 4.1. Random Bit Sequence Generator

An LFSR is used to generate a random bit sequence. This random sequence in the proposed design will act as an input to the convolutional encoder.

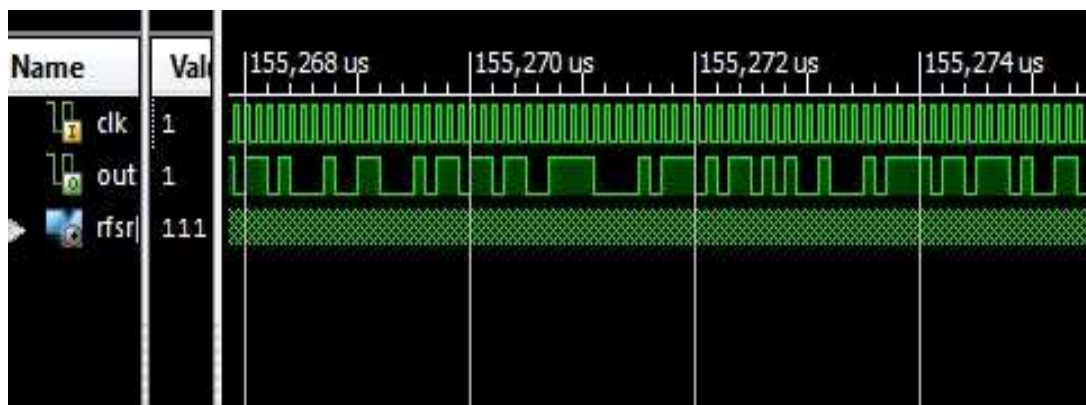


Figure 4.1: LFSR implementation generating random sequence

## 4.2. Convolutional Encoder

The theory of convolutional encoder has already been explained in the last chapter. In the proposed design, the following convolutional encoder has been used. The input to the encoder is random bit sequence which is taken from the LFSR.

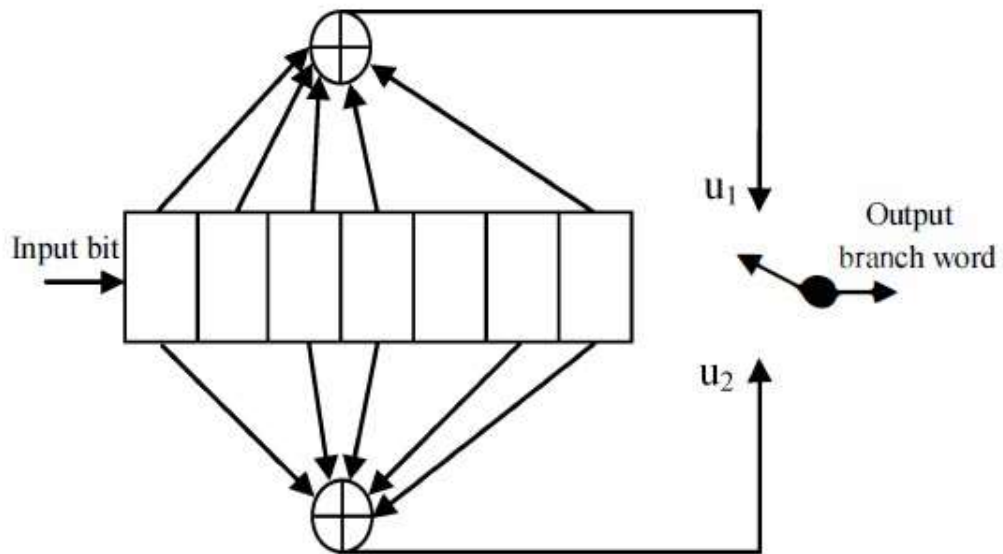


Figure 4.2: Convolutional Encoder

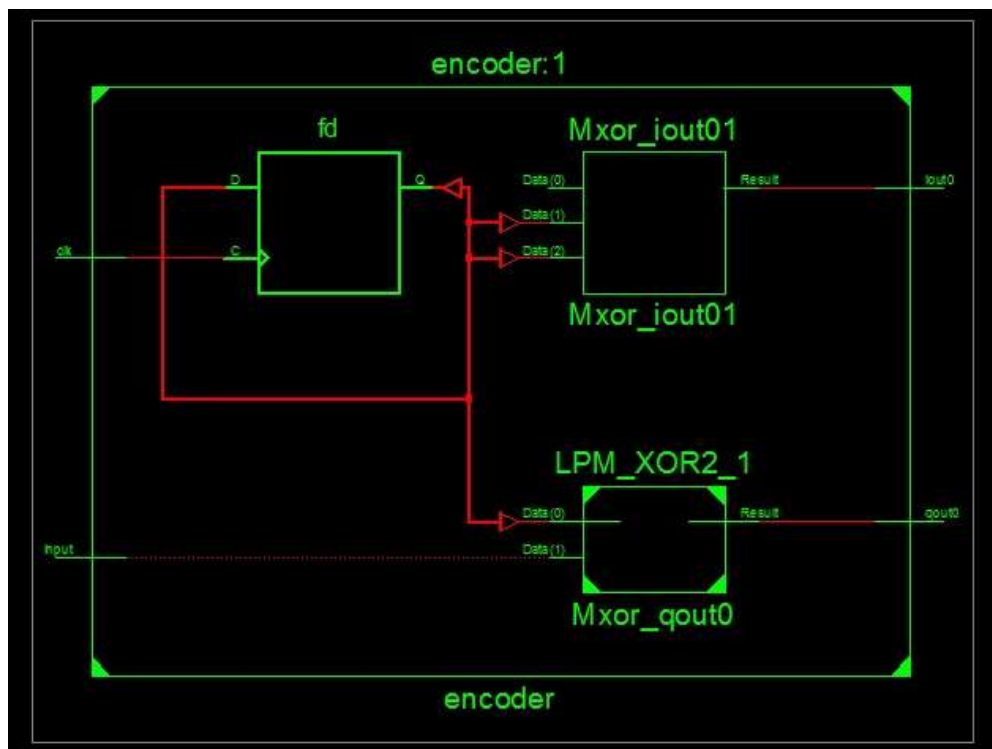


Figure 4.3: Schematic Diagram of Proposed Encoder

### 4.3. QPSK Modulator

The convolutional encoded bits are modulated using QPSK modulation. QPSK modulator is having RRC filter in it. The designing of RRC filter is done using MATLAB. The designing of RRC filter has been done for roll-off factor 0, 1 and 2. The convolutional encoded bits are encoded using these 3 roll-off factors.

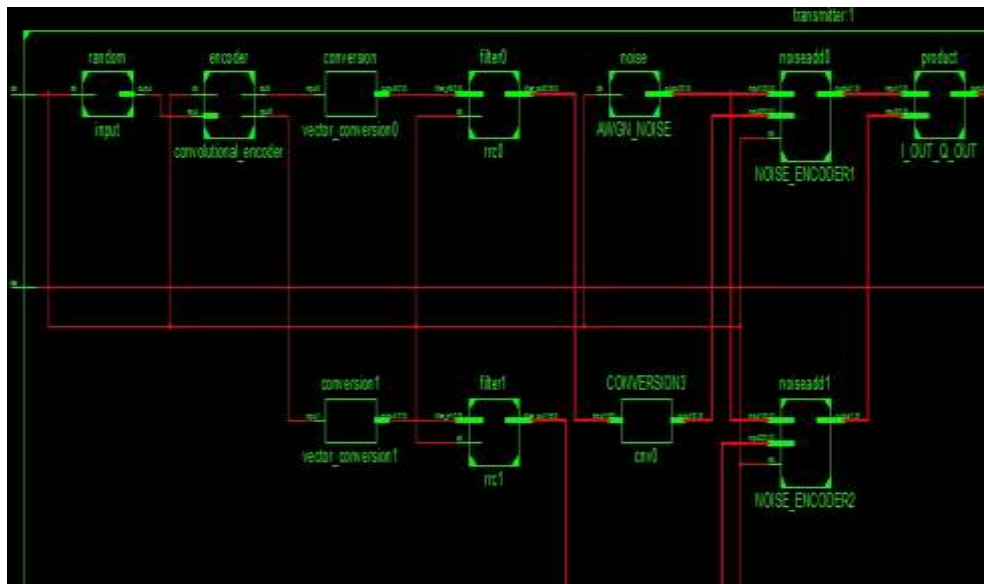


Figure 4.4: Schematic of QPSK modulator with convolutional encoder

### 4.4. AWGN Channel

AWGN is a basic model used to calculate the effect of Random processes occurred in nature.

Additive: - added to any noise.

White: - uniform emissions for all frequencies in visible spectrum.

Gaussian: - average time domain value is zero in the time domain normal distribution.

In wireless communication, AWGN (Additive White Gaussian Noise) channel introduces most of the noise in real data. It is actually a mathematical model that represents physical phenomena in which the impairment is the linear addition of white noise with a constant spectral density.

To simulate AWGN channel in VHDL, two random functions have been generated which gives two variables. Then by using Box-Muller method, the noise signal is calculated. [21], [22], [23], [24]



#### 4.6. Final Proposed Design

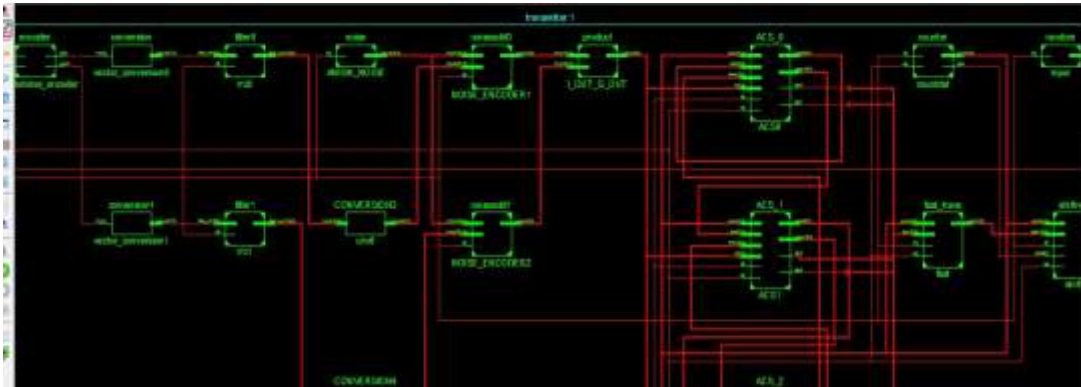


Figure 4.8: Schematic of final proposed design

As the QPSK modulation scheme is being used in this proposed design, QPSK is having RRC filter in it. In RRC, roll-off factor is one of the parameter used. So, by considering the different roll-off factors, bit error per 100 clock pulse has been calculated and graphs has been plotted to check with which roll-off factor one can get best performance.

❖ Roll-off factor = 0

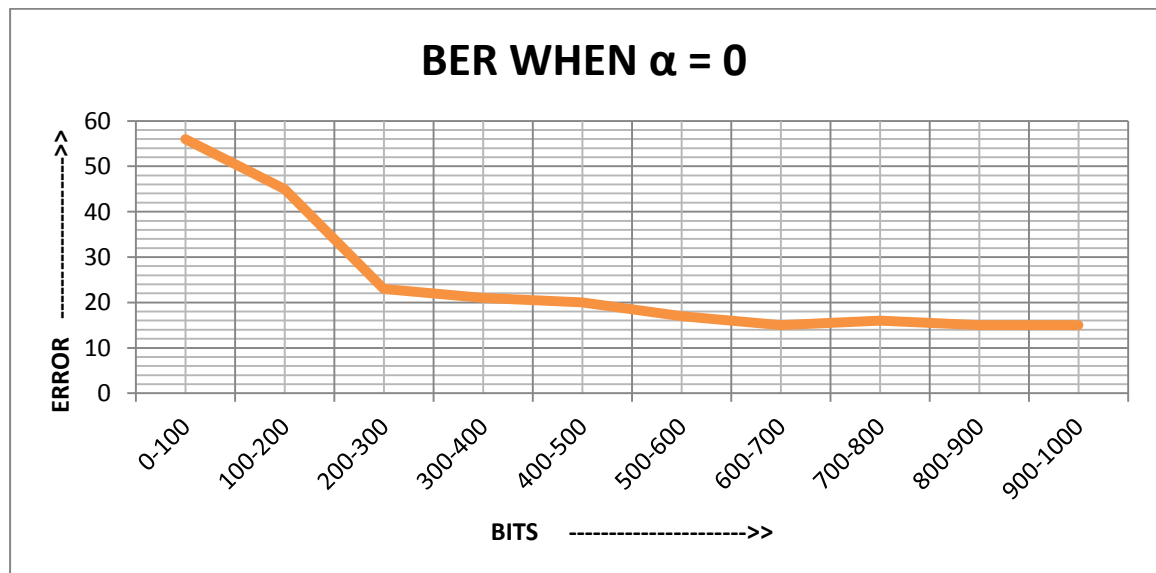


Figure 4.9: Graph BER when  $\alpha = 0$

The BER curve is decreasing due to increase in the logical AND operations as the number of bits are increasing.

❖ Roll-off factor = 1

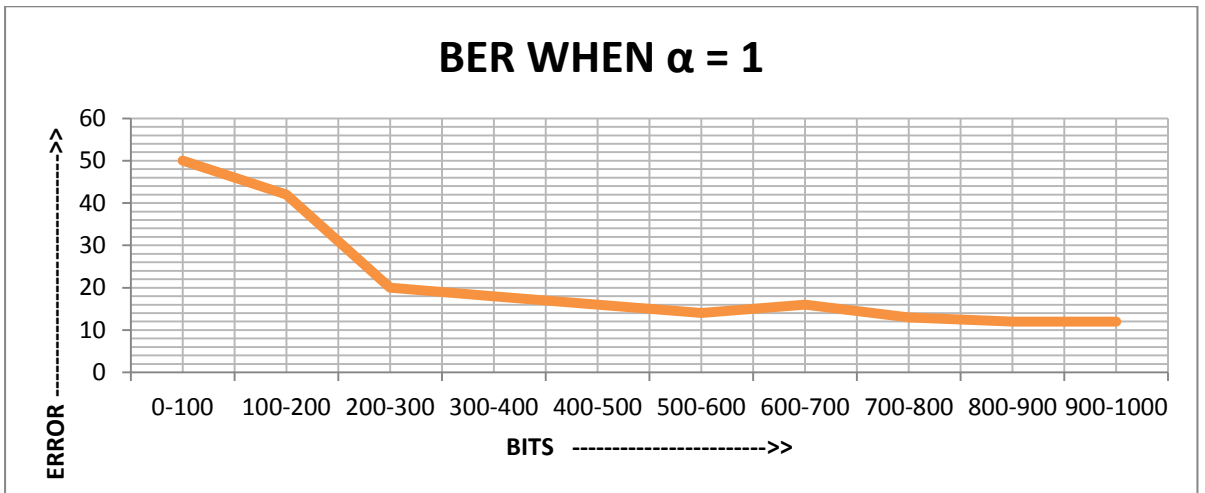


Figure 4.10: Graph BER when  $\alpha = 1$

❖ Roll-off factor = 2

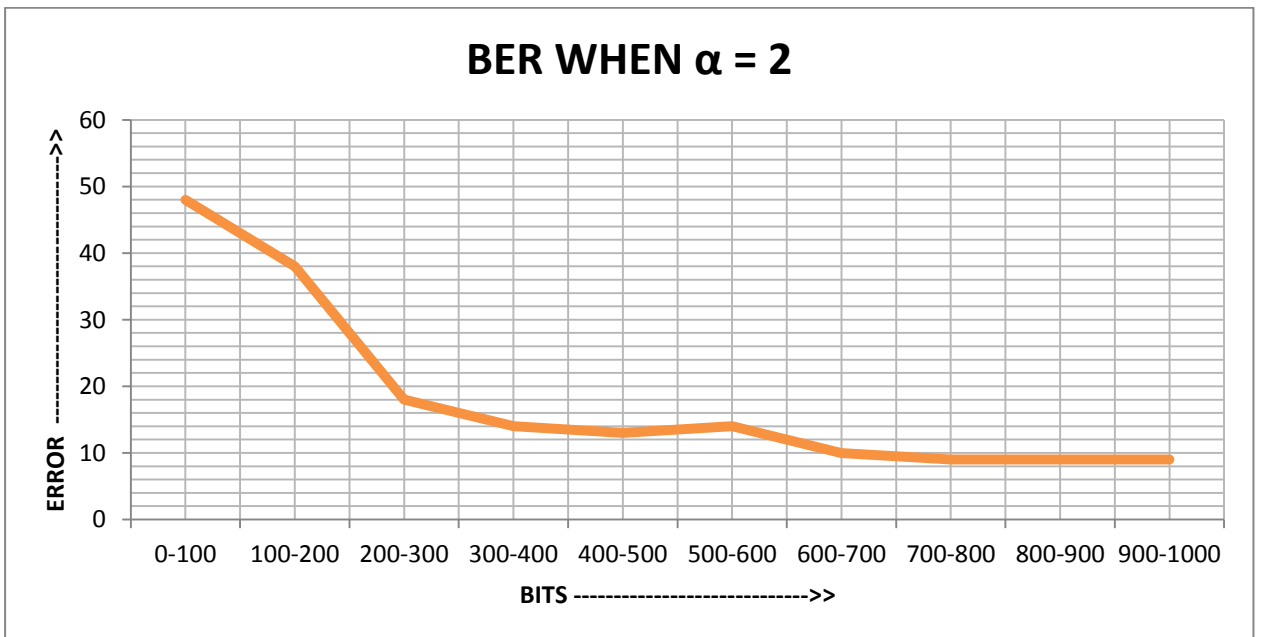


Figure 4.11: Graph BER when  $\alpha = 2$

As the number of bits is increasing and as the roll-off factor is increased from 0 to 1 and 1 to 2, BER decreases, this is due to the increase in the complexity and number of AND operations in the RRC filter as roll-off factors increases.

## Comparison

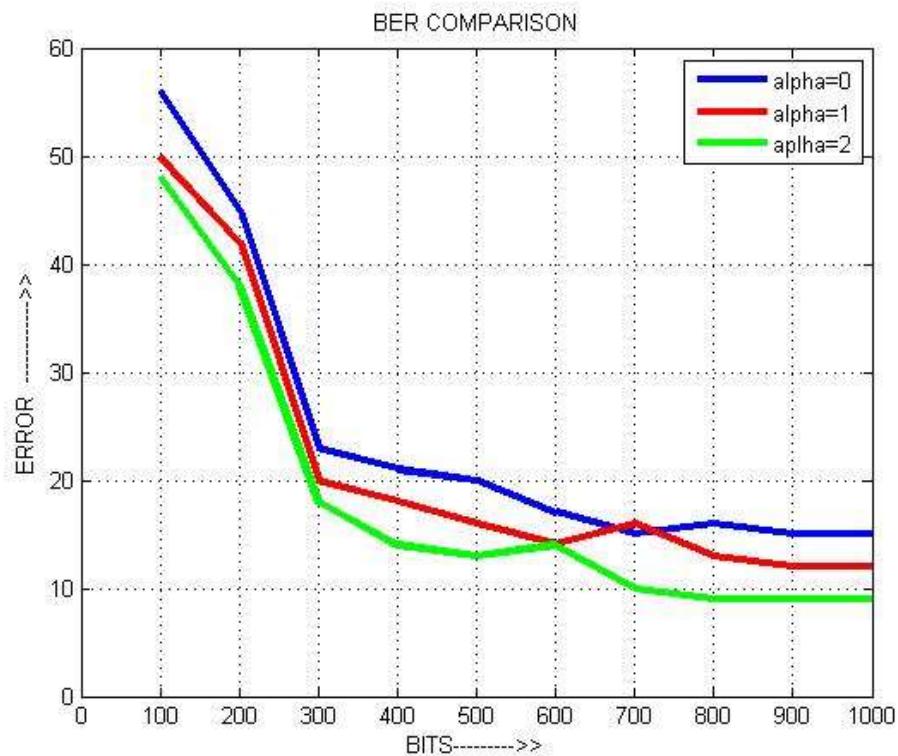


Figure 4.12: Comparison among roll-off factors

After seeing the individual plotted graph and comparison graph (figure 4.12), it becomes very clear that by using RRC filter with roll-off factor = 2, error in bits decreases as the bit rate increases. During the first 100 clock pulse, the BER is approximately 52-53 for all three roll-off factors. But during 900-1000 clock pulse BER decreases to a great extent and it is around 10 for roll-off factor 2.

```

Device utilization summary:
-----

Selected Device : 3s500epq208-5

Number of Slices:                489 out of 4656 10%
Number of Slice Flip Flops:      395 out of 9312  4%
Number of 4 input LUTs:          685 out of 9312  7%
  Number used as logic:           677
  Number used as Shift registers:  8
Number of IOs:                    4
Number of bonded IOBs:            4 out of 158  2%
Number of GCLKs:                  1 out of 24   4%

```

Figure 4.13 Device utilization summary for proposed design

This chart (figure 4.13) shows that only 10% devices of the total devices are used in the proposed device which makes this proposed design a minimum resource using design. Whereas all the previous architectures discussed in the literature survey has more than 15% device utilization.

```

=====
Final Register Report

Macro Statistics
# Registers                               : 388
  Flip-Flops                               : 388
# Shift Registers                          : 7
  2-bit shift register                     : 4
  29-bit shift register                    : 1
  6-bit shift register                     : 1
  7-bit shift register                     : 1
=====

```

Figure 4.14 Final Register Report

Figure 4.14 represents the final report for synthesis. This says that 7 shift registers are used in this proposed design. Whereas no. of flip-flop used are 388. All the previous architecture uses around 500 registers in all.

By making comparison among proposed design and previous architectures it is clear that the proposed design is having less no. of device utilization.

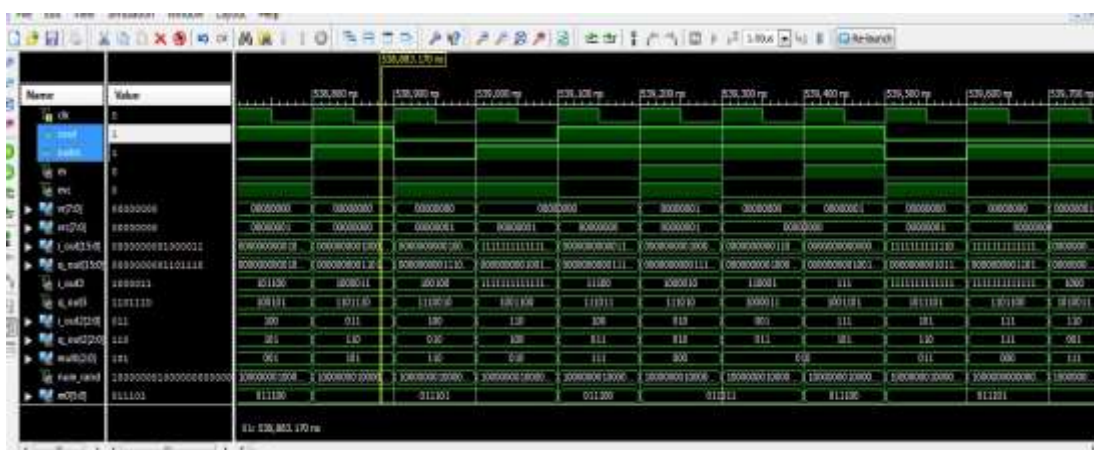


Figure 4.15 Final Output

Figure 4.15 shows that clock pulse is continuous; sout is the final output from the decoder whereas outin is the input to the encoder. Rests are the signals which are used to implement the proposed design.

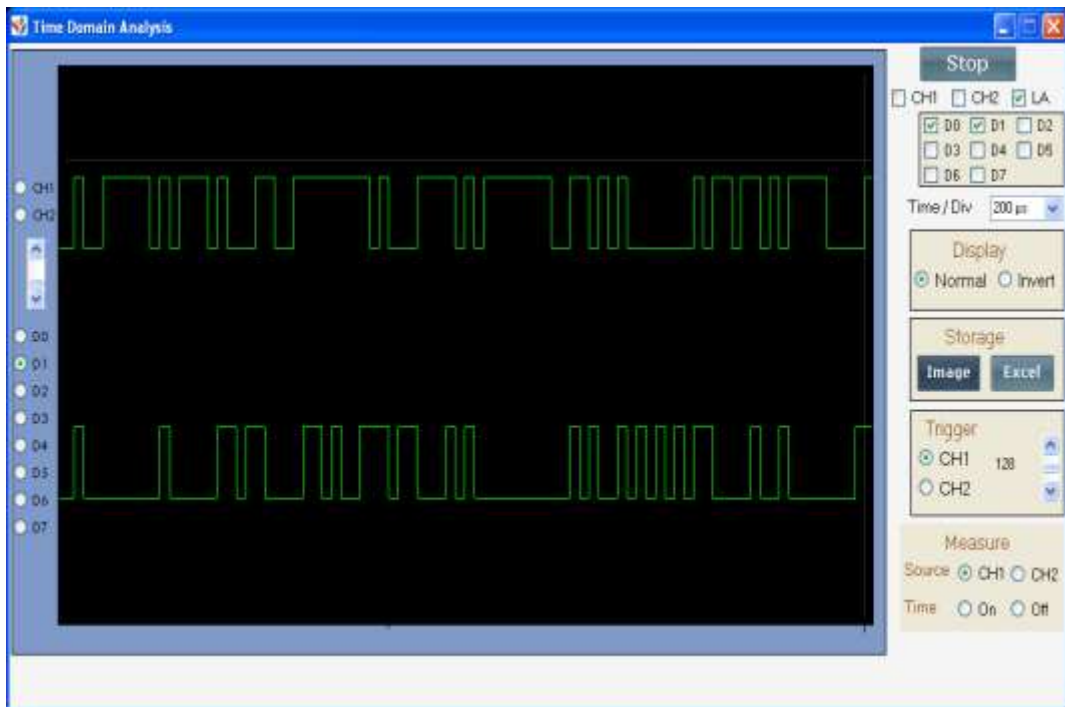


Figure 4.16: SDR implementation of convolutional encoder and Viterbi decoder

### Timing and Power Summary

After the synthesis report, the timing diagram generated according to the given input. With the help of timing diagram speed grade, Minimum period, Maximum Frequency, Maximum output required time after clock is calculated.

#### Timing Summary

- Speed Grade: -4
- Minimum period: 27.362ns
- Maximum Frequency: 36.547MHz
- Minimum input arrival time before clock: 4.213ns
- Maximum output required time after clock: 4.182ns

#### Power summary

Total estimated power consumption: P (mw): 42 mw

## 5. CONCLUSION AND FUTURE SCOPE

### 5.1 Conclusion

In this thesis, firstly random bit sequence has been generated with the help of LFSR. Then this random bit sequence is encoded with the help of convolutional encoder. The encoded bits are then modulated over three roll-off factors value of 0, 1 and 2 with the help of RRC filter in QPSK modulation technique. These modulated bits are then decoded with the help of Viterbi decoder. The channel through which receiver is receiving the transmitted bits is AWGN channel. This thesis also presents the design of AWGN channel. The proposed design has described the SDR implementation of Convolutional Encoder and Viterbi Decoder over AWGN channel. Using QPSK modulation RRC filters got used and depending upon the various roll-off factors BER over every 100 bits got analyzed. The timing summary after analysis got is minimum period is 27.362ns (i.e. maximum frequency is 36.547 MHz). Maximum input arrival time before clock is 4.213ns and maximum output required time after clock is 4.182ns.

### 5.2 Future Scope

This proposed design can be implemented using

- Fading Channels
- Different coding techniques
- For different Filters

## 6. REFERENCES

- [1] Sajjad Ahmed Ghauri, Hasan Humayun, Muhammad Ehsan ul Haq, Farhan Sohail, "Implementation of Convolutional Codes on FPGA," *7<sup>th</sup> International Conference for Internet Technology and Secured Transactions*, Volume. 08, Page No. 175-178, Dec 2012.
- [2] Pravallika. kolakaluri, R. Suryaprakash, B. Vijay Bhaskar, "HDL Implementation of Convolution Encoder and Viterbi Decoder", *International Journal of Engineering Research & Technology*, Volume.1, Issue 5, Page No. 1-5, July 2012.
- [3] Nejah Nasri, Laurent Andrieux, Abdennaceur Kachouri, Mounir Samet, "Behavioral Modeling and Simulation of Underwater Channel", *WSEAS Transactions on Communications*, Issue 2, Volume. 8, Page No. 259-268, Feb 2009.
- [4] J.Tulasi, T.Venkata Lakshmi, M.Kamaraju, "FPGA Implementation of Convolutional Encoder and Hard Decision Viterbi Decoder", *International Journal of Computer & Communication Technology*, Issue 4, Volume 3, Page No. 43-47, Dec 2012.
- [5] Yan Sun, Zhizhong Ding, "FPGA Design and Implementation of a Convolutional Encoder and a Viterbi Decoder Based on 802.11a for OFDM", *Wireless Engineering and Technology*, Volume.3, Page No. 125-131, July 2012.
- [6] Jinjin He, Huaping Ling, Zhongfeng Wang, Xinming Huang and Kai Zhang, "High-Speed Low-Power Viterbi Decoder Design for TCM Decoders", *IEEE Trans. VLSI*, Volume. 20, Page No. 755-759, April 2012.
- [7] Rohan M. Pednekar, Dayanand B M., "Design and Implementation of Convolution Encoder with Viterbi Decoder", *International Journal of Emerging Technologies in Computational and Applied Sciences*, Volume 6, Issue 1, Page No. 84-89, Sept 2013.
- [8] Bineeta Soreng, Surabh Kumar, "Efficient Implementation of Convolutional Encoder and Viterbi decoder", *International Conference on circuits, power and computing technologies*, Page No. 1270-1273, Oct.2013.

- [9] Yin Sweet Wong, Wen Jian Ong, Jin Hui Chong, Chee Kyun Ng, Nor Kamariah Noordin, "Implementation of Convolutional Encoder and Viterbi Decoder using VHDL", *Proceedings of 2009 Student Conference on Research and Development*, Volume 4, Page No. 22-25, Nov 2009.
- [10] Mahe Jabeen, Salma Khan, "Design of Convolution Encoder and Reconfigurable Viterbi Decoder", *International Journal of Engineering and Science*, Volume 1, Issue 3, Page No. 15-21, Sept 2012.
- [11] Yan Sun, Zhizhong Ding, "FPGA Design and Implementation of a Convolutional Encoder and a Viterbi Decoder Based on 802.11a for OFDM", *Wireless Engineering and Technology Scientific Research*, Volume 2, Page No. 125-131, July 2012.
- [12] J.Tulasi, T.Venkata Lakshmi, M.Kamaraju, "FPGA Implementation of Convolutional Encoder and Hard Decision Viterbi Decoder", *International Journal of Computer & Communication Technology*, Issue 4, Volume 3, Page No. 43-47, Dec 2012.
- [13] Hiral Pujara, Pankaj Prajapati, "RTL Implementation of Viterbi using VHDL", *IOSR Journal of VLSI and Signal Processing*, Volume 2, Issue 1, Page No. 65-71, April 2013.
- [14] Sherif Welsen Shaker, Salwa Hussien Elramly, Khaled Ali Shehata, "FPGA Implementation of a Reconfigurable Viterbi Decoder for WiMAX Receiver", *International Conference on Microelectronics*, Volume 1, Page No. 258-261, July 2009.
- [15] Dayanand B M., M. Sida, U. Ahola, and D. Wallner. "Bluetooth transceiver design and simulation with VHDL-AMS," *IEEE Circuits and Devices Magazine*, Volume 19, Page No 11-14, March 2003.
- [16] P. Nikitin. E. Nomark C. Wakayama., R. Shi, "VHDL-AMS madding and simulation of a BPSK transceiver system" *Proceeding of IEEE International Conference on Circuits and Systems for Communications*, 2004.
- [17] C.Arun, Swati Gupta, Rajesh Mehra, "FPGA Implementation Of Viterbi Decoder using Trace Back Architecture", *International Journal Of Engineering Trends and Technology*, Page No. 131-134, June 2011.
- [18] Irfan Habib, Özgün Paker, Sergei Sawitzki, "Design Space Exploration of Hard-Decision Viterbi Decoding: Algorithm and VLSI Implementation" *IEEE*

*Tran. on Very Large Scale Integration (VLSI) Systems*, Volume. 18, Page No. 794-807, May 2010.

- [19] Kling, "Channel Coding Application for CDMA2000 implemented in a FPGA with a Soft processor Core," Linköping University, Department of Electrical Engineering, 2005.
- [20] S. Hema, V. S. Babu and P. Ramesh, "FPGA Implementation of Viterbi Decoder," *Proceedings of the 6th WSEAS Int. Conf. on Electronics, Hardware, Wireless and Optical Communications*, Corfu Island, Greece, February 16-19, 2007.
- [21] L. Dong, "Cross-layer design for cooperative wireless networks," *Drexel Thesis and Dissertations*, 4 Dec. 2008.
- [22] P. Subhashini, D. R. Mahesh Varma and Y. David Solomon Raju, "Implementation Analysis of adaptive Viterbi Decoder for High Speed Applications", *International Journal of Computer Applications*, Volume 31, Issue.2, Page No. 0975—8887, October 2011.
- [23] S.V.Viraktamath and G.V.Attimarad, "Impact of constraint length on performance of Convolutional Codec in AWGN channel for image application", *International Journal of Engineering Science and Technology*, Volume 2, Issue 9, Page No. 4696-4700, Dec 2010.
- [24] Ramtin, I. OudinoL S . Seatti. A. Le-Clery, and I. Lebm., "Full transceiver Circuit simulation using VHDL-AMS" *International Conference of Microwave Engineering*, Page No. 29-33. May 2002.
- [25] A. Fakhfakh, N. Milet - his , I. Tomas, and H. Levi, "Behavioral modeling of phase noise and jitter in voltage-controlled oscillators with VHDL-AMS." *Proceedings of IEEE International Conference on Circuits and Systems for Communication*, Page No. 370-373, 2002.
- [26] U. Knoehel, Hang . and R. Kakerow, "Verification of the RF subsystem within wireless LAN system level simulation" *Design, Automation and Design in Euorpe Conference*. Page No. 286-291, 2003.

## LIST OF PUBLICATIONS

Published:-

Abhishek Aggarwal, Dr. Rajesh Khanna, “SDR Implementation of Convolutional encoder and Viterbi decoder”, *International Journal of Advanced Research in Electronics and Instrumentation Engineering*, Volume 3, Issue 5, Page No. 9571-9576, May, 2014.