DESIGN AND PERFORMANCE ANALYSIS OF POLAR CODES

THESIS SUBMITTED

IN PARTIAL FULFILLMENT OF THE REQUIREMENT

FOR THE AWARD OF THE DEGREE OF

MASTER OF ENGINEERING

In

ELECTRONICS AND TELECOMMUNICATION ENGINEERING

By

PRONAB PAUL

EXAMINATION ROLL No: M4ETC19002

REGISTRATION No: 140693 of 2017-18

Under the guidance of

DR. JAYDEB BHAUMIK

DEPARTMENT OF ELECTRONICS & TELECOMMUNICATION ENGINEERING

JADAVPUR UNIVERSITY, KOLKATA-700032

WEST BENGAL, INDIA

MAY-2019

DECLARATION

I hereby submit this thesis for partial fulfillment of MASTER OF ENGINEERING IN ELECTRONICS AND TELECOMMUNICATION ENGINEERING degree. I also declare that this thesis contains original work and related literature survey. All information in this document have been obtained and presented in accordance with academic rules and ethical conduct. I also declare that as required by these rules and conduct I have fully cited and referenced all materials and results that are not original with this work.

NAME: PRONAB PAUL

EXAMINATION ROLL NO: M4ETC19002

REGISTRATION NO: 140693 of 2017-18

PROJECT TITLE:

DESIGN AND PERFORMANCE ANALYSIS OF POLAR CODES

DATE:

PLACE: _____

(PRONAB PAUL)

Faculty of Engineering & Technology

Jadavpur University

CERTIFICATE OF RECOMMENDATION

This is to certify that the thesis entitled **"DESIGN AND PERFORMANCE ANALYSIS OF POLAR CODES"** has been carried out by **Pronab Paul** (Examination Roll No: M4ETC19002 and Registration No:140693 of 2017-18) under my guidance and supervision and be accepted in partial fulfillment of the requirement for the degree of Master of Engineering in Electronics & Telecommunication Engineering.

Countersigned by:

Dr. Jaydeb Bhaumik
Thesis Supervisor
Dept. of ETCE
Jadavpur University
Kolkata-700032
West Bengal, India

Prof. Dr. Sheli Sinha Chaudhuri
Head, Dept. of ETCE,
Jadavpur University,
Kolkata 700032
West Bengal, India

Prof. Chiranjib Bhattacharjee
Dean, FET
Jadavpur University, Kolkata
West Bengal, India

Faculty of Engineering & Technology

Jadavpur University

CERTIFICATE OF APPROVAL*

The forgoing thesis, entitled **"DESIGN AND PERFORMANCE ANALYSIS OF POLAR CODES"** is hereby approved as a creditable study on an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or accept every statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it has been submitted.

Committee on Final Examination for
Evaluation of the Thesis:

External Examiner

Dr. Jaydeb Bhaumik
SUPERVISOR

*Only in case the thesis is approved.

ACKNOWLEDGEMENT

This is a great opportunity to express my gratitude and respect to all the persons who have supported me, each, in his/her own way, during my Master course.

It is difficult to overstate my gratitude to my respected Supervisor, Dr. Jaydeb Bhaumik, for guiding and inspiring me throughout my thesis period at Jadavpur University. I highly appreciate him for making me understand what research is. Moreover, I will always remain indebted to him for encouraging and giving me so much time, in spite of his busy schedule.

I would like to thank Prof. Dr. Sheli Sinha Chaudhuri, H.O.D, Dept. of ETCE, Jadavpur University for providing me all the facilities for carrying out the entire project work.

I would also like to thank our Industrial Lab members for their support and inputs during my research.

A special thanks to my friend, Prapty Paul, without whose help I couldn't complete my project. I would also like to acknowledge all my class-mates and supporting staffs for their support and motivation.

Last but not the least, I would like to thank my parents Mr. and Mrs. Paul, and other family members for their support and motivation.

May 2019

ABSTRACT

Polar codes, invented by Arıkan in 2008, is one of a channel coding technique that provably achieve Shannon capacity on any binary-input discrete memory less channel (BI-DMC). Polar codes have been selected as a control channel coding technique in the 5^{th} generation (5G) wireless communication systems. These codes are suitable for practical applications mainly due to their low complexity and recursive structure for both encoding and decoding. Polar codes perform better than LDPC codes when decoded with advanced list successive cancellation decoder with cyclic redundancy check (LSCD with CRC), which is a significant improvement of the original successive cancellation decoder (SCD) proposed by Arıkan.

The whole work in this thesis is divided into two parts. In the first part, we have shown FPGA-based implementation of Successive Cancellation Decoder (SCD) for short code length (N= 8, 16, 32, 64) by reducing the numbers of XOR operations to make the design compact. Here we have used Verilog Hardware Description Language (Verilog-HDL) for implementation of each module. By reducing the numbers of XOR operations, we have achieved up to 70% improvement in area overhead. Synthesis results are presented in the first part.

In the second part, we have shown some performance analysis of Successive Cancellation Decoder (SCD) in terms of BER vs. Eb/N_o and FER vs. Eb/N_o for different code lengths, different code rates and for systematic and non-systematic codes.

We have also shown some performance analysis of CRC-Aided Successive Cancellation decoder in terms of bit error rate vs. SNR and block error rate vs. SNR for different code rates, different List size and different modulation scheme.

CONTENTS

List of Figures

List of Tables

Chapter 1

Introduction

The purpose of digital communication systems is to provide reliability to the noisy channel during transmission or storage of digital data. The basic block diagram of a digital communication system is shown in Figure 1.1. Entire procedure is divided into mainly encoding, modulation, demodulation and decoding. Source encoder removes the redundant information from the source's data before sending to channel encoder. Channel encoder adds redundancy to the data such that reliable communication can be taken place over a noisy channel. Before transmission, modulation of data is taken place. Demodulator recovers data from modulated data bits. Function of the channel decoder is to reproduce the data sent over the channel from the demodulator. At the receiver end, source decoder reproduces the source's data from the output of channel decoder. Our main objective in this thesis is the design and performance of polar decoder.

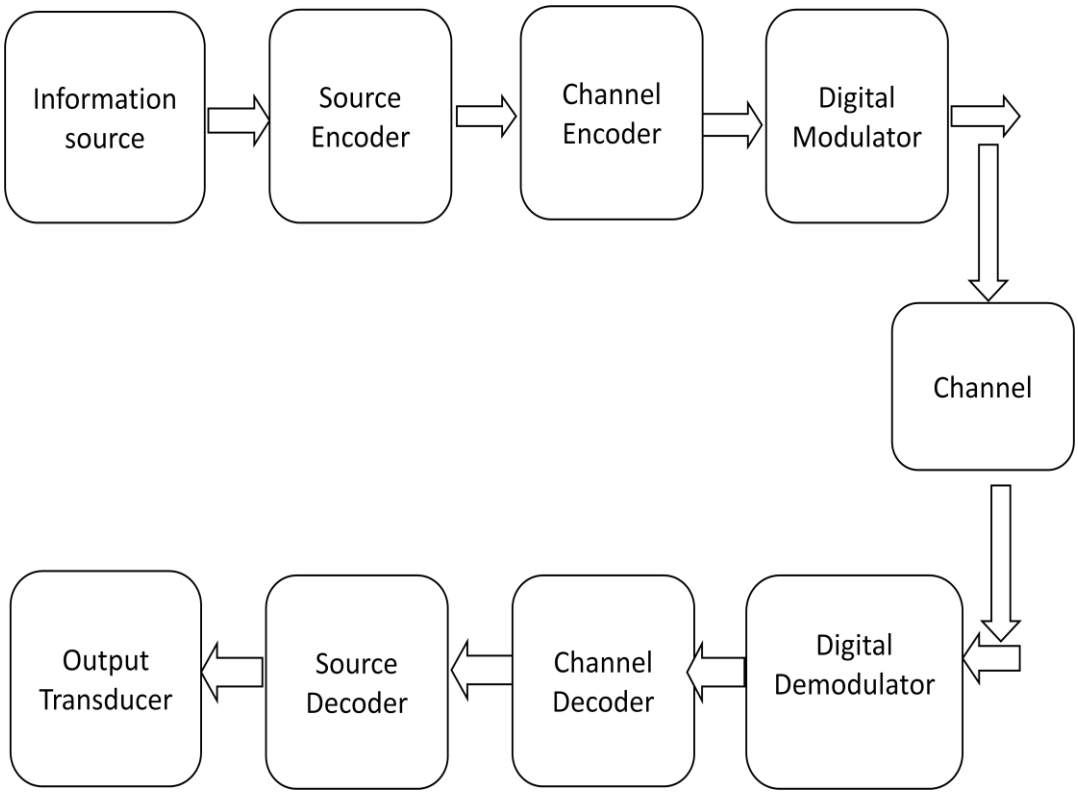


Figure 1.1: Basic Block diagram of a digital communication system

Nearly seventy years ago, Claude Shannon set a limit: No matter how advanced our technology becomes, how sophisticated our algorithms are, we cannot reliably transmit information at a rate, or efficiency, higher than what he called the channel capacity. However, he left the question of how to achieve this channel capacity, is unanswered. For decades, researchers are trying to find

methods to achieve, or even approach, the channel capacity. These efforts were successful in 1993 when turbo codes were discovered .These codes came close to achieve capacity in certain cases and re-energized the field of error correcting codes. Low density parity-check (LDPC) codes, discovered by Gallager in the 1960swere rediscovered and found to have excellent error-correction performance also.

In 2008, Arıkan invented polar codes [1], from then total game changed for the race of achieving Shannon limit. Polar codes, which asymptotically achieve the symmetric capacity on any binary input discrete memory less channels (BI-DMC). More importantly, polar codes achieve symmetric capacity with an explicit non-random construction, using the low-complexity successive-cancellation decoding algorithm where Turbo and LDPC code require randomness in their construction, complicating that process and increasing routing complexity in hardware implementations. Polar codes exploit the channel polarization phenomenon, in which certain bits are always estimated reliably while others are completely unreliable when decoding using successive cancellation.

1.1 Motivation

Transferring information between devices plays an important role today. The rising demand for error free data transmission and the planning of 5G communication increases the interest in finding ways to improve the rate and reliability of data transmission. Channel coding plays an important role for improving these; they reduce the number of errors occurring on channels to any desirable level.

Recently significant research works are being performed on polar codes which are an important Shannon capacity approaching channel codes. It is proven that polar codes can achieve channel capacity for long block lengths. Recursive and low complexity decoder structure make them suitable for hardware implementation. Therefore, Polar codes have received much attention nowadays. However short block length polar codes of up to a few thousand bits show relatively poor performance and are hence not commonly used in practice without modification. It is found that codes that work well for short block lengths are hard to construct. The low complexity and excellent performance of polar codes for long block lengths have motivated lots of research, trying to improve them also for short block lengths.

Researchers found a way to improve the performance of polar decoder, using the combination of Successive Cancellation List (SCL) decoding and Cyclic Redundancy Check (CRC) code. The SCL decoder extends the Successive Cancellation (SC) decoder that was originally used with polar codes. It gives us a list of possible codewords when decoding, and the CRC check is employed in the last decoding step to select the correct codewords from the list. These added elements introduce little complexity but have shown impressive results for short block lengths. Polar codes with this combination go from relatively poor performing to competing with well-researched codes for short block lengths. These codes are even the best performing codes for

some channels. Another recent development is that polar codes have been selected as the standard to implement in 5G for short block lengths.

It is now interesting to find out how the design complexity of polar decoder can be reduced or how can we improve the performance of decoder by sacrificing some other parameters, or what is the specific combination of polar codes, list decoding and CRC that creates better codes?

The structural difference between the codes lies in the selection of frozen bits, which in short means what set of possible codewords can be sent on the channel. How does the selection of frozen bits influence the improvement caused by SCD and CRC? What is the CRC size needed to full fill the necessity of the polar code? These are the questions which motivated us to work on polar code and in this thesis we have tried to find answer of some of these questions.

1.2 Related Work

The interest of this thesis grew from previous related works on polar codes, their similarity with RM (Reed Muller) codes, and results showing, polar codes with list decoding and CRC perform better than other codes available today for short block lengths on some channels.

Arikan first introduced polar codes in 2008 [1], where they were shown to achieve channel capacity for infinite block lengths. Already in that article which Arikan published, the close association between polar codes and the older RM codes was mentioned, and the relation was further explained by Arikan in [2]. A performance comparison between the two codes, without adding outer codes and for short block lengths of 32 and 256 bits, was shown in [3], finding polar codes to perform better than RM codes for short block lengths under SC decoding. They were also compared in [4] using Maximum Likelihood (ML) decoding, under which the RM code performed better than polar codes, but at a high cost of decoding complexity.

Tal and Vardy suggested the use of list decoding with CRC on polar codes [5] and their first results were shown in [6]. The article [6] includes results, showing improved polar codes of 2048-bit block lengths beating low density parity check (LDPC) codes of length 2304 and rate 0.5. The improved polar code combination was also looked at in [7], showing polar codes of the short block length of 128 bits and rate 0.5 to perform better than both LDPC and turbo codes of the same size and code rate, for frame error rates down to10^{-6}.

Some proposals for improving the selection of frozen bits for polar codes have been made. Vangala and Viterbo compared different suggestions of selecting frozen bits in [8] for (2048, 1024) codes. It was shown in [9] that Log Likelihood Ratio (LLR) values of unfrozen bits could be used to identify unreliable channels and swap them with some good frozen channels, which improve the polar code performance. Experiments have also been done with combining polar codes and Reed-Muller codes in different ways in [10] and [11], finding better and improved results of these interpolated/hybrid Reed Muller-Polar codes.

Reed-Muller codes were invented in 1950's but have only recently been proven to reach capacity. These codes, being older than polar codes, have been the subject of interest for a lot of

research through the years, including many suggested improvements of the decoder. Recursive list decoding and improvement of the frozen bit selection for short length Reed-Muller codes was suggested in [12].

Many studies have also been made, trying to improve the decoding technique for Successive Cancellation Decoder and recently work is going on to find the best cyclic redundancy check (CRC) polynomials. Search for best polynomials that can be used for different block lengths for long burst error detection, is described in [13], suggesting a good practical way of selecting general CRC polynomials for short codes.

Improvement in Hardware design of polar codes is now a matter of study. Lot of works have been done on FPGA design for SCD based Polar Decoder to improve the performance. In [14] an efficient SC polar decoder based on new folding approaches is proposed to achieve very low processing complexity with proper combinations of decomposition method and folding technique.

A high-throughput energy efficient SCD architecture based on combinational logic is proposed in [15]. The suggested architecture operates at relatively low frequencies compared to high frequencies of sequential circuits, but takes advantage of the high degree of parallelism inherent in such architectures to provide a favorable tradeoff between throughput and energy efficiency at short to medium block lengths. The advantage of using combinational logic is that it has low complexity which helps to get high throughput.

1.3 Thesis Contribution

Previous works on polar codes helped us to understand the polar codes. In this thesis, FPGA implementation of Successive Cancellation Decoder for short code length is shown. Verilog Hardware Description Language (Verilog-HDL) is used for implementation of each module. We have used different code length (N=8, 16, 32, 64) for FPGA implementation of SCD. We have tried to make the design compact by reducing the numbers of XOR operations compared to traditional implementation. By reducing the numbers of XOR operations, we have achieved up to 70% improvement in area overhead.

- Some analysis of SCD in terms of Bit Error Rate (BER) vs. Eb/N_o and Frame Error Rate (FER) vs. Eb/N_o has been performed for different code lengths and different code rates.

- Also performance analysis of CRC-Aided SCD has been presented in terms of bit error rate vs. SNR and block error rate vs. SNR for different code rates, different list sizes and different modulation scheme. Performance analysis of SCD and CRC-Aided SCD has been performed using MATLAB tool.

- FPGA implementation of SCD has been performed employing Xilinx ISE Design suite 14.7 software.

1.4 Thesis Outline

This thesis work contains of eight chapters. The remainder of the thesis is organized as follows.

Chapter 1: Chapter 1 is all about invention and development of polar codes in terms of achieving Shannon limit.

Chapter 2: This chapter is about channel polarization and construction of polar codes.

Chapter 3: Chapter 3 is regarding encoding and decoding of polar codes. It also discuss about importance of Bhattacharyya parameter in encoding.

Chapter 4: This chapter discuss about construction and properties of Successive Cancellation Decoder.

Chapter 5: Chapter 5 is about CRC aided successive cancellation decoding. This chapter also discuss about CRC aided SCL decoding.

Chapter 6: In this chapter FPGA implementation of SCD for short length code word is discussed. Implementation output and design summery is also given here.

Chapter 7: Chapter 7 is all about performance analysis of SCD and CRC aided SCD in terms of bit error rate, frame error rate and block error rate with Eb/No and SNR.

Chapter 8: Chapter 8 concludes the thesis work with future scope.

Chapter 2
Polar Codes

The purpose of channel coding is to find codes which can be transmitted quickly, contain many valid codewords and can correct or at least detect many errors. Performance in these areas is a trade-off between accuracy and complexity. Different codes are used for different applications. The needed properties of this code mainly depend on the probability of errors happening during transmission. The general idea for error detection and correction is to add some redundancy to a message, which receivers can use to check consistency of the delivered message, and to recover data that has been determined to be corrupted. Error-detection and correction schemes can be either systematic or non-systematic: In a systematic scheme, the transmitter sends the original data, and attaches a fixed number of check bits (or parity data), which are derived from the data bits by some deterministic algorithm. If only error detection is required, a receiver can simply apply the same algorithm to the received data bits and compare its output with the received check bits; if the values are not same, an error has arrived at some point during the transmission. In a system that uses a non-systematic code, the original message is transformed into an encoded message that has at least as many bits as the original message.

2.1 Introduction to polar codes

In the field of error correcting codes, a modern channel coding technique has developed known as polar coding and it is a technique which was discovered by Erdal Arıkan in 2008[1]. Polar codes are considered to gain channel capacity for a given binary input discrete memory less channel (BI-DMC). This can be happened only when the code length is sufficiently large. Polar codes are under testing for a long time and are at last going to be used by Huawei in 5^{th} generation networks by 2020 [16].

A Polar code is said to be a linear block error correcting code. Various concatenations that are recursive is said to be the fundamental structure for the polar code. And this is the basis for the code construction [25]. Physical transformation of the channel takes place, which transforms the physical channels to virtual channels and this transformation is based on multiple concatenations that is recursive.

2.2 Polar codes channel model

Before proceeding with the polar codes, it is very important to know the construction and properties of channel W, [1] which is a binary discrete memory less channel (BDMC).Ultimate objective of selecting good channels are to achieve symmetric capacity *I(W),*which is the highest rate of data transmission possible in communication system.

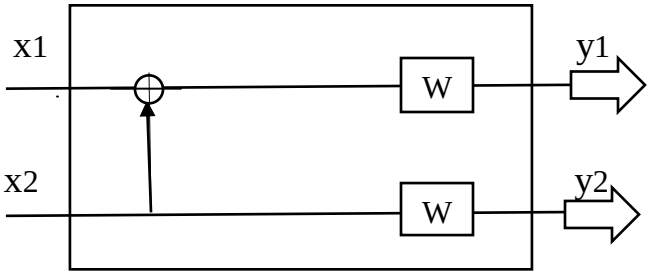


Figure 2.1: Polar code basic Channel

It is possible to synthesize or create a second set of N binary input channels from the original N channels. The binary input channels have the properties $\{W_N^{(i)}: 1 \leq i \leq N\}$. For polar codingchannel model, it is important to remember that when numbers of input channels (N) increase, two possibilities are about to occur i.e. transmission rate for some fractions of channels for I ($W_N^{(i)}$) is near to 0; in terms of capacity achieving it can be represented as 1- I(w). Transmission rate for rest of the fractions of channels for I($W_N^{(i)}$) is about 1; that means these fractions of channels are approaching towards capacity achieving I(w).These polarized channels I{$W_N^{(i)}$} are in good condition for channel coding. So the channels which can transmit at a rate of 1 or almost 1 are used for polar coding. Some channels which have transmission rate about to 0 are called junk channels.

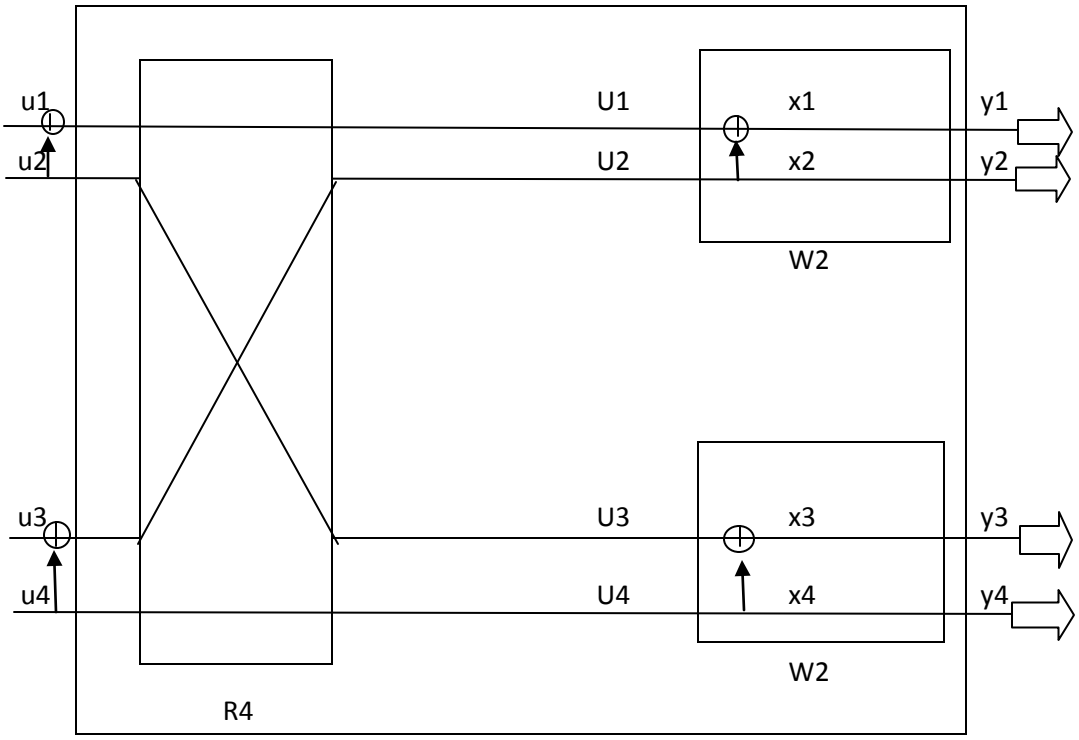
For a binary discrete memory less channel W, there are basically two parameters of interest. One is the symmetric capacity I(W) and the second one is the Bhattacharya parameter [17].

$$I(W) = \sum_{y \epsilon Y} \sum_{x \epsilon X^{\frac{1}{2}}} w\left(\frac{y}{x}\right) log\left(\frac{w\left(\frac{y}{x}\right)}{0.5\,(w(y|0) + w(y|1))}\right.$$

And the Bhattacharyya parameter is given by

$$Z(w) = \sum_{y \in Y} \sqrt{W(y|0)W(y|1)}$$

The symmetric capacity parameter is used to measure the rate and the Bhattacharya parameter is considered as a parameter of reliability. Symmetric capacity is the highest rate at which any reliable communication can be taken place across the channel W. Bhattacharya parameter Z(W) is said to be an upper bound on the probability of maximum-likelihood (ML) decision error when W is used only once to transmit either of the two i.e. 0 or 1.



W_4

Figure 2.2: Channel W4 and its relationship with W and W2 [1]

In polar code, channel combining model is very important to observe. Channel combining process of polar codes are shown in figure 2.2 where two copies of W are constructing $W_2$and combination of two W_2 are creating W_4. As the number of channels increases the number of stages for combination also increases at a rate 2^n.

In polar coding, generally three standard notations are used to observe the behaviour of various functions. These notations are known as Landau notations, and represented as $O(N)$, $o(N)$, $\omega(N)$ [1]. Certain notations are used to denote the realizations and some for probability assignment. For example, the random variable P and Q will be denoting their realizations or sample values and the lower-case letters p and q will be used to specify their probability assignment. According to the notation, PP specifies the probability assignment on P. Similarly for both P and Q joint probability can be written as P (P, Q). The mutual information and its conditional form can be denoted as i.e. I (P; Q), I (P; Q |Z) respectively.

Standard Landau Notations $O(N)$, $o(N)$, $\omega(N)$ denotes the asymptotic behaviour [2] of various functions. Some standard function notations which are commonly used in polar coding channel model are as follows

Let, P and Q are random variables

p, q - realizations of random variables (RV) of P and Q respectively.

PP - probability assignment on a random variable P.

P (P, Q) – joint probability assignment for a join ensemble of random variables P and Q

I (P; Q), I(P; Q |Z) – denotes the mutual information and it's conditional form respectively

A row vector $(a_1, \ldots, a_N)$ will be denoted by a_1^N

$a_i^j \leq i,J \leq N, 1 \leq i, j \leq N$, to denote the sub vector $(a_i, \ldots, a_j)$; if j<i, a_i^j is regarded as void.

a_1^j, can be used to denote the sub vector with even indices (a^k: $1 \leq k \leq j$; k even)

The notationO_1^N can be used to denote the all zero vector.

2.2.1 Channel Polarization

Channel polarization can be defined as a process by which N independent copies of a given binary input discrete memory less channel (BI-DMC) W creates a second set of N channels $\{WN^{(i)} : 1 \leq I \leq N\}$ that show polarization effect. Channel polarization consists of two phases, known as channel splitting and channel combining. In channel splitting phase W is divided into N copies of new channels (W_1,W_2,....,W_N). Similarly in channel combining phase N copies of independent channel create a new channel W.

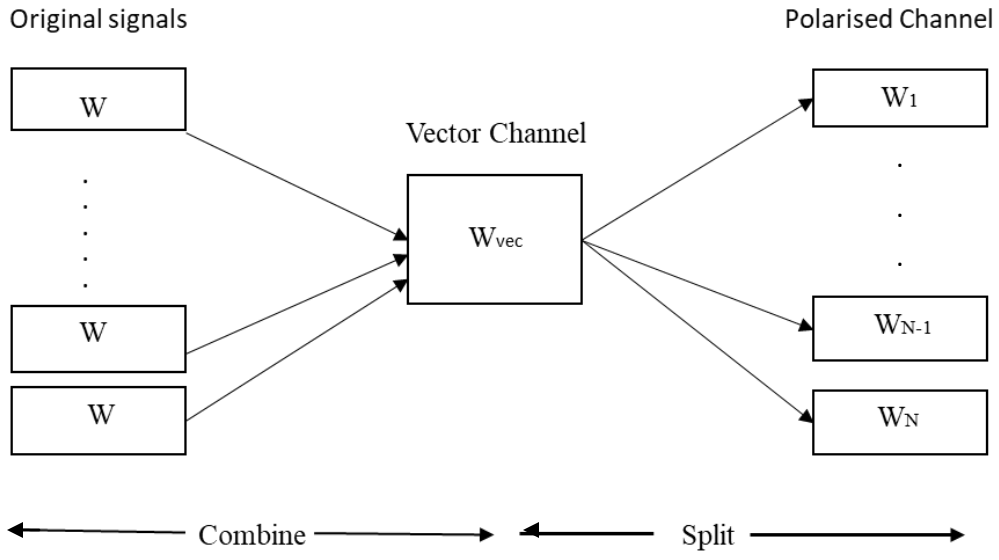


Figure 2.3: Channel Polarization: combining and splitting

The theorem of polarization is stated below. It is more related to the concept of source polarization. As it is seen from the Theorem below, when the bit-channel capacities polarize the number of channels either move towards 1 or 0. It is considered that, for fraction of terms for which conditional entropy lies between δ and 1- δ. For instance, consider δ to be something like 1%. In this case, one would expect the interval between δ and 1- δ to be large because it occupies almost all the interval. As N goes towards Infinity, the terms between δ and 1- δ goes to 0 so there is almost nothing which means there are almost no channels in the middle, and all have shifted to either the lower end 0 or upper end 1[1].

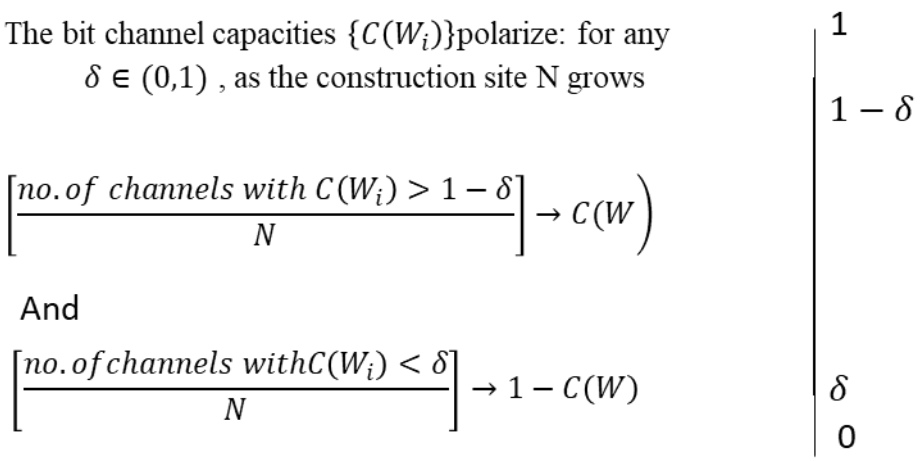


Figure 2.4: Channel polarization depicting 1- δ and δ [1]

2.3 Code Construction

Polar codes use the channel polarization method to achieve the symmetric capacity of a Binary memory less channel (BMC) as the code length increases to infinity. A simple construction of polar code where N= 2 is shown in Figure: 2.5a. The probability of correctly estimating bit u_1 increases compared to when the bits are transmitted without any transformation over the channel W. Meanwhile, the probability of correctly estimating bit u_0 decreases. The polarizing transformation can be combined recursively to create longer codes, as shown in Figure: 2.5b for N= 4. As the code size increase to infinity, the probability of successfully estimating each bit approaches either 1 (fully reliable) or 0.5 (totally unreliable), and the proportion of reliable bits approaches the symmetric capacity of W [1].

To construct an (N, K) polar code, where N is code length and K information length and(N–K) is the least reliable bits, these (N-K) bits are called the frozen bits, which are set to zero and the remaining K bits are used to carry information. The locations of the information bits K and frozen bits (N-K) are based on the type and conditions of W unless specified otherwise. In this thesis we use polar codes constructed according to [18] [19].

The generator matrix, G_N, for a polar code of length N can be specified recursively so that $G_N=F_N= F2\bigoplus^{logN}$, where F2 = $\begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}$and $\bigoplus$is the kronecker power. For example, for N=4 GN is

$$G_4 = F_2^{\otimes 2}=\begin{bmatrix}F_2 & 0\\ F_2 & F_2\end{bmatrix}=\begin{bmatrix}1 & 0 & 0 & 0\\ 1 & 1 & 0 & 0\\ 1 & 0 & 1 & 0\\ 1 & 1 & 1 & 1\end{bmatrix}$$

In matrix form, non-systematic encoding can be represented as $x=u.G_N$, where u is a N-bit row vector containing the bits to be encoded in the information bit locations. When polar codes were initially proposed, bit-reversed indexing was used. While this changes the bit ordering for both encoding and decoding, the error-correction performance remains unaffected. This change translates into multiplying the generator matrix by the bit-reversal permutation matrix B_N [1] (or $\tilde{\Pi}_N$[20]), so that $G_N = B_NF_N$.

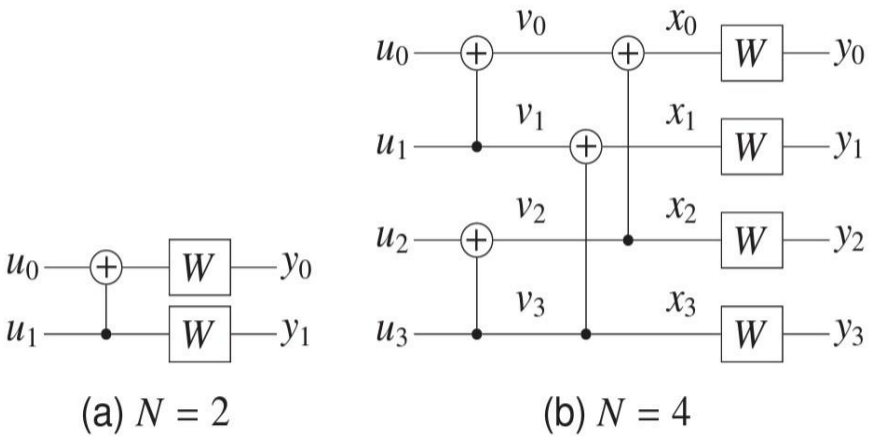


Figure 2.5: Construction of polar codes of lengths 2 and 4

Chapter 3

Encoding & Decoding

3.1 Encoding

Encoding is the process to convert data into a suitable format for transmission and storage. Polar encoding can be done by using simple linear mapping. For the code block length L the generator matrix, G_L is defined as $G_L = B_L F^{\oplus n}$ for any $L = 2^n$ as $n \geq 1$, where B_L is a bit-reversal matrix and $F^{\oplus n}$ is the n^{th} kronecker power of the matrix

$$F=\begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}$$

Polar encoder with code length L has an input vector $u^L{}_1$ and an output vector $y^L{}_1$.The mapping of $u \rightarrow y$ is linear over the binary field F^2such that,$y^L{}_1 = u^L{}_1 G_L$, where G_L is the generator matrix.

The factor graph representation of 8-bit encoder is shown in Figure.3.1, where $\oplus$ symbol represents binary XOR operation.

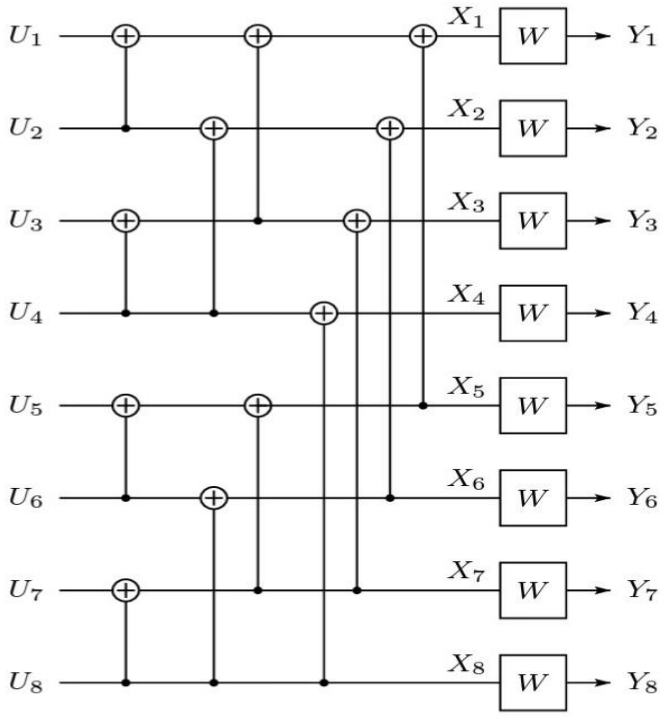


Figure 3.1: The factor graph representation of 8-bit encoder

3.1.1 Non Systematic Encoding

Binary polar codes are generally specified by a triple (L, A, I), where L is the code length, A is the length of the message, and $I \subseteq L$, $|I| = A$ is the set of information bit indices. The remaining $L-A$ bits are called as frozen bit indices. Here, L is a power of 2 and we define $n = \log_2(L)$. For a (L, A, I) polar code, the generator matrix is $G = F^{\otimes n} I$. A non-systematic [20] codeword x is represented as:

$$x = u \cdot G_L = d \cdot F^{\otimes n} I$$

where u is a message vector of A information bits and d is a vector of L bits including information bits such that $d_I = u$ and $d_{jc} = 0$. The bits d_{jc} are called as frozen bits, and are set to zero. Due to the recursive construction of the matrix $F^{\otimes n}$, this matrix-vector multiplication can be performed in (N log N) only. This non-systematic polar encoder is illustrated in Figure: 3.2. Non-systematic encoding is the original encoding process for polar codes.

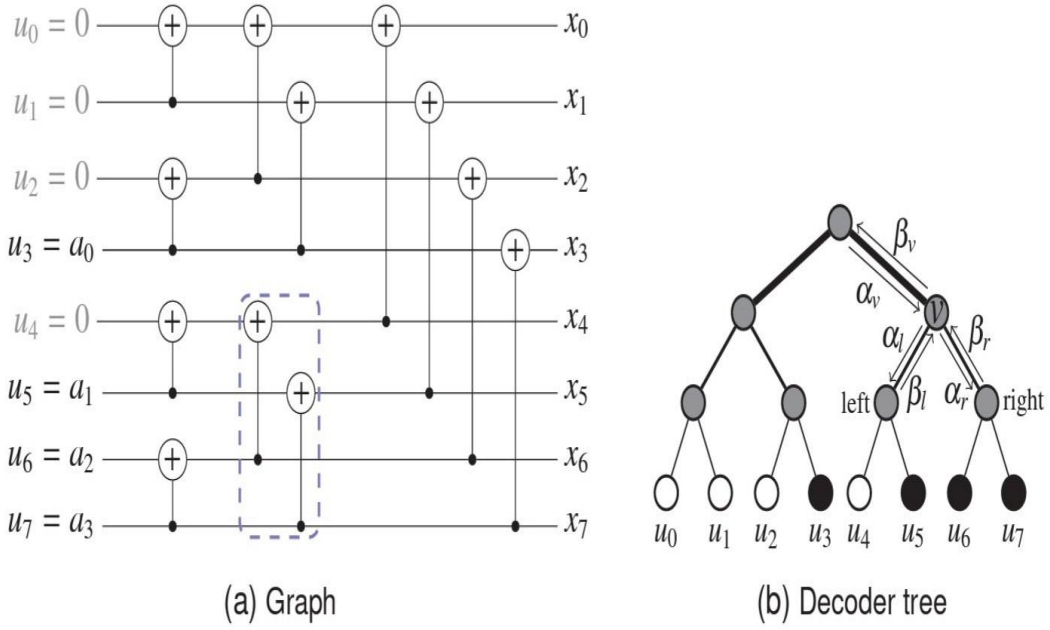


Figure 3.2: Non-systematic (8,4) polar code represented as a (a) graph and as a (b) decoder tree

3.1.2 Systematic Encoding

A systematic polar code may be considered as equivalent to the original polar code (non-systematic), except that the message vectors are mapped to code words. For a (L, A, I) code word it is considered that the bit indices of A in a code word **x**, appear explicitly. It was shown in [20] that bit indices of A can be chosen equal to the set of information bit indices I. It is

important to notice that it slightly differs from what is usually considered as a systematic linear block code, where message bits appear as first A bits.

Figure: 3.3 shows an example of the low-complexity systematic encoding scheme. It comprises two non-systematic encoding passes and a bit masking operation in between. For a (8, 4) polar code, a L-bit vector u= [0, 0, 0, a_0, 0, a_1, a_2, a_3], where a_0, ….,a_3 are the A= 4 information bits, enters the first non-systematic encoder from the left. Then, using bit masking, the locations corresponding to frozen bits are reset to 0 before propagating the updated vector through the second non-systematic encoder. The end result is an L-bit vector. Finally the encoded code word is expressed as x = [p_0, p_1, p_2, a_0, p_3, a_1, a_2, a_3], where p_0, ….,p_3 are the L−A = 4 parity bits and a_0, ...,a_3 are the information bits.

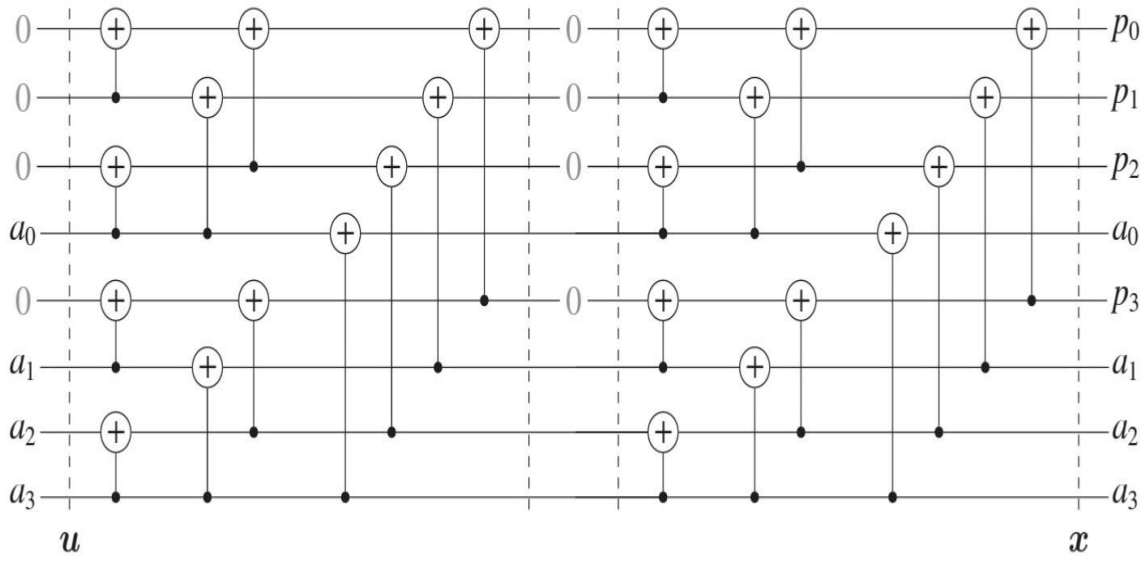


Figure 3.3: Low-complexity systematic encoding of a (8, 4) polar code.

Systematic polar codes offer better bit error rate (BER) [21] than their non-systematic counterparts, while frame error rate (FER) is also improved slightly. Furthermore, systematic encoding allows the use of low-complexity rate-adaptation techniques such as code shortening method proposed in [22]. Flexible low-complexity systematic encoding of polar codes is discussed in [23].

3.2 Frozen Bits, and their Selection

Bhattacharyya parameter was used to select frozen bits in the construction of polar codes in [1]. Some other frozen bit computations have been suggested and used since then. A comparison between some different suggested ways of finding frozen bits for polar codes is shown in [18]. In our thesis, Bhattacharyya parameter is used when computing channel reliabilities and selecting frozen bits.

The Bhattacharyya parameter measures the reliability of channels. It is an upper bound of the probability that an error can occurs using maximum-likelihood decision when a channel W is used to transmit a single bit. The different physical channels have different Bhattacharyya parameter, and design of polar codes depends on the physical channel. Bhattacharyya parameter is represented as:

$$Z(W) = e^{-Ec/No}$$

For polar code construction an iterative algorithm is used. The process of iteration uses two equations (a).

$$Z(w^{-}) \leq 2Z(W) - Z(W^2)$$
$$Z(W^{+}) = Z(W^2) \qquad \text{(a)}$$

Iteration process is shown in Figure 3.4 where it is notable that the process continues until the tree has N leaves. All leaves give the new channel Bhattacharyya values for all N polarized channels. N channel values are found after n = log_2(N) iterations. Every step in the tree uses the equations on the current Z-value to find two child node Z-values. The N−k channels with the highest Bhattacharyya values are selected as frozen channels [1]. Frozen bit values are known to both the sender and receiver, and are always set to zero.

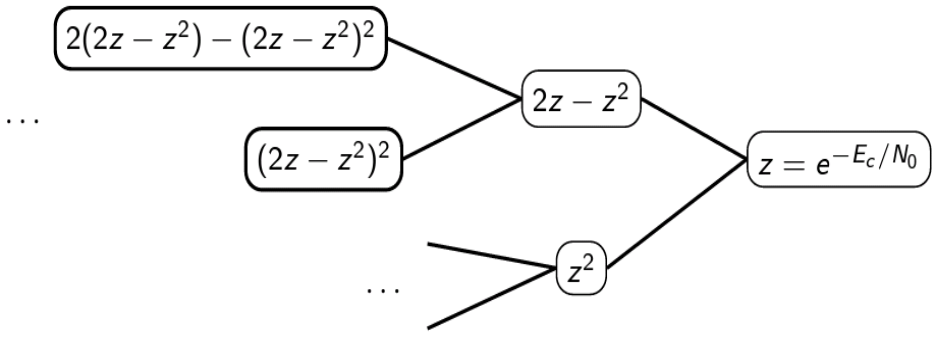


Figure 3.4: Bhattacharyya parameters calculation on polar channels

Polar codes adapt to channels, so a polar code found for one channel might not be the polar code found for another. This is due to how the Bhattacharyya parameter of the physical channel influences the values of all polarized channels. It is not practical to change the set of frozen bits for a polar code after implementation; hence it is usually optimized for a special design-SNR. The polar code is designed for one channel with some channel parameters, and keep the same set of frozen bits if the channel changes.

3.3 Decoding

Polar codes are generally decoded using an improved low complexity sequential algorithm [24] called as successive cancellation decoding (SCD). Undoubtedly, the SCD for its simplicity became the prime reason for all the glory of polar codes after having been established as capacity achieving by Arıkan. Even today, most of the practical decoders such as the list decoder are variants of SCD.

When Encoding is done, information bits are combined with parity bits. Like if information is k and parity is p then after encoding code word is C= k+p. Decoding is the process to get back information bits. SCD is the basic building block of polar decoding. Base on SCD lots of advanced and efficient Decoding scheme has been introduced, like CRC aided successive cancellation list decoding. In the next chapters we will be discussing about Successive Cancellation Decoder and CRC Aided SCD in details. A butterfly representation of SCD is shown in figure 3.5

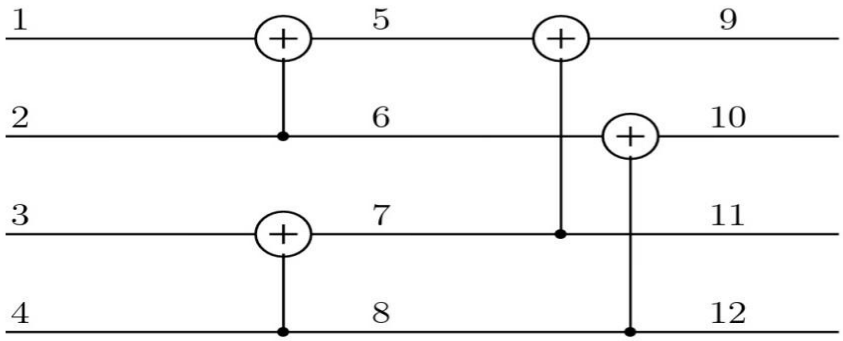


Figure 3.5: Polar Decoder for N=4

Chapter 4

Successive Cancellation Decoding

Successive Cancellation Decoder is the basic building block of decoding of polar codes. Nowadays, lots of modified and advance decoding technique is available, and they are performing well, but Successive cancellation decoding still an important part in polar coding.

4.1 Successive Cancellation Decoder

Just as the name implies, a Successive Cancellation decoder means decoding the bits in order from $u_1$to u_N, this also helps to realize channel polarization. As it is stated before, the mutual information I(u_i; $y^N{}_1$, $u^{i-1}{}_1$) or Bhattacharyya parameter Z(u_i; $y^N{}_1$, $u^{i-1}{}_1$) is required to predict the channel performance. Then the decoder should has the knowledge of $u^{i-1}{}_1$ when decoding u_i. Generally, the decoder knows the values of frozen bits $\{u_j, j \in Ac\}$ in advance only. As the Successive cancellation decoder [26] decodes bits consecutively, it at least provides an estimate of $u^{i-1}{}_1$ when decodingu_i. Arikan has shown that if the information set A is properly specified, this has no impact on the error performance if the code length is large enough.

Consider a polar code with the parameter vector (N, K, A, uA_c). The decoder needs to retrieve the information and generate an estimate $\hat{u}^N{}_1$ of $u^N{}_1$ with the knowledge of $y^N{}_1$, A and uA_c. The likelihood ratio (LR) is defined as [1].

$$L_N^{(i)}(y_1^N, u_1^{i-1}) = \frac{W_N^{(i)}(y_1^N, u_1^{i-1}|u_i = 0)}{W_N^{(i)}(y_1^N, u_1^{i-1}|u_i = 1)}$$

For convenience, the frozen bits are always set to 0. Then a Successive Cancellation decoder would take the following steps to achieve translating the source bits. For each i from 1 to N:

- If i$\in A_c$, $u^i = u_i$.
- If I $\in$A, calculate the LR $L^{(i)}{}_N(y^N{}_1, u^{i-1}{}_1)$ and make the decision as

$$\hat{u}_i = \begin{cases} 0, & \text{if } L_N^{(i)}(y_1^N, u_1^{i-1}) \geq 1 \\ 1, & \text{otherwise} \end{cases}$$

The LR can be recursively calculated as [1]

$$L_N^{2i-1}\left(y_1^N, \hat{u}_1^{2i-2}\right) = \frac{L_{N/2}^{(i)}\left(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}\right) \cdot L_{N/2}^{(i)}\left(y_{N/2+1}^{N}, \hat{u}_{1,e}^{2i-2}\right) + 1}{L_{N/2}^{(i)}\left(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}\right) + L_{N/2}^{(i)}\left(y_{N/2+1}^{N}, \hat{u}_{1,e}^{2i-2}\right)}$$

$$L_N^{2i}\left(y_1^N, \hat{u}_1^{2i-1}\right) = \left[L_{N/2}^{(i)}\left(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}\right)\right]^{1-2\hat{u}_{2i-1}} L_{N/2}^{(i)}\left(y_{N/2+1}^{N}, \hat{u}_{1,e}^{2i-2}\right)$$

The recursive calculation of LRs can be traced back to code length 1 with the LR $L^{(i)}1\ (y_i) = \frac{W(y_i|0)}{W(y_i|1)}$. $L^{(1)}1\ (y_i)$ is the soft information observed from the channel.

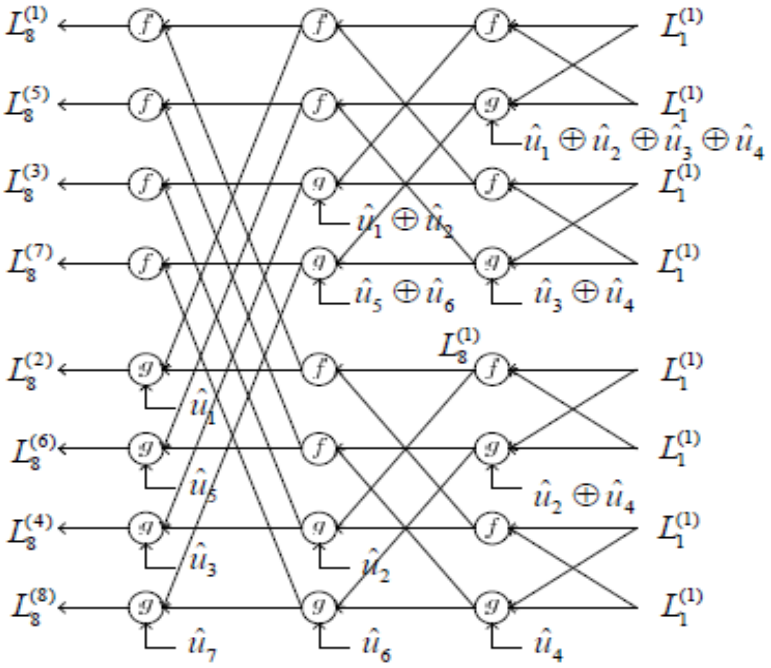


Figure 4.1: N=8, SC polar decoder architecture [31]

Figure 4.1 shows the implementation of a Successive Cancellation decoder with code length N=8. The calculation of $L^{(i)}N$ is conducted from right to left. In this figure, f and g represent two functions as follows

$$\begin{cases} f(a,b) = \dfrac{1+ab}{a+b} \\ g(a,b,\hat{u}_s) = a^{1-2\hat{u}_s} b \end{cases}$$

The variables a and b denote $L^{(i)}N/2(y^{N/2}1,\ u\hat{}^{2i-2}1{,}0 \oplus u\hat{}^{2i-2}1{,}e\ )$ and $L^{(i)}N/2(y^{N}N/2{+}1,\ u\hat{}^{+2i-2}1{,}e\ )$, respectively.

Algorithm: Successive Cancellation Decoding

Input: Received Codeword, x_1^N

Input: code block length, N

Input: Information set, k

Input: frozen bit vector, u_{k^c}

Output: estimated free bits, $\hat{u}_k$

begin

1. **for** i ⟵ 1 to N do
2. **if** i ∉ k then
3. $\hat{u}_i \longleftarrow u_i$
4. **else**
5. **if** $\log\left(\frac{W_N^{(i)}(x_1^N, \hat{u}_1^{i-1} | \hat{u}_i = 0)}{W_N^{(i)}(x_1^N, \hat{u}_1^{i-1} | \hat{u}_i = 1)}\right) \geq 0$ **then**
6. $\hat{u}_i \leftarrow 0$
7. **else**
8. $\hat{u}_i \leftarrow 1$
9. **return** $\hat{u}_A$

4.2 Successive Cancellation List Decoding

Successive Cancellation List decoding algorithm [27] is accepted to enhance the performance of Successive Cancellation decoding for short and moderate block lengths in [6]. SCL decoding enables tracking L best decoding paths concurrently, unlike a SC decoder can track at most a single decoding path. If L is sufficiently large, Maximum Likelihood decoding performance is achieved, since sufficient numbers of decoding paths are visited. A trade-off between complexity and performance of the algorithm is going on, because time complexity (γ_{SCLD}) and space complexity (δ_{SCLD}) of the algorithm linearly depends on list size (L) such that

$$\gamma_{(SCLD)}(L, N) = O\ (L\ N \log N)$$
$$\delta_{(SCLD)}(L,N) = O\ (L\ N)$$

A high level description of the algorithm is shown in Algorithm 2. The Successive Cancellation List decoding algorithm takes the received codeword y^N_1, the code block length N, the information set k, the frozen bit vector u_{kc} and the maximum list size L as input and calculates the estimated information bits u^k as output. The current list size variable, cL set as 1 at the initialization of the algorithm. If the i^{th} hard decision belongs to the frozen set k^c, the i^{th} hard decisions of all L lists are updated with the frozen decision, u_i In case of a free decision, the decoder checks whether the current list size is equal to the maximum list size. If they are not equal, the current list size doubles and the decoder can track likelihoods of both decisions. In case of all lists are occupied, the decoder sorts 2L likelihoods to continue with the best L decoding paths. At the end of the last decision step, the decoder outputs the free bits from the best list as u^k.

Algorithm: Successive Cancellation List Decoding

Input: Received Codeword, x_1^N

Input: code block length, N

Input: Information set, k

Input: frozen bit vector, u_{k^c}

Input: maximum list size, L

Output: estimated free bits, $\hat{u}_k$

Variable: cL ← 1//current list size

1. begin
2. for i ⟵ 1 to N do
3. **if** i ∉ A then
4. for l ← 1 to cL do
5. $\hat{u}_{l,i} \leftarrow \hat{u}_i$
6. **else**
7. **if** cL ≠ L **then**
8. for l ← 1 to cL do
9. $\hat{u}_{l,i} \leftarrow 0$
10. $\hat{u}_{l+cL,i} \leftarrow 1$
11. $cL \leftarrow 2_{cL}$
12. **else**
13. $s \leftarrow \text{sort}\left(W_N^{(i)}\left(x_1^N, \hat{u}_1^{i-1} \middle| \hat{u}_{1,i}^L\right)\right)$
14. **for** l ← 1 to cL do
15. $\hat{u}_{l,i} \leftarrow s_l$
16. **Return** $\hat{u}_k$

Chapter 5

CRC-Aided Successive Cancellation Decoding

The choice of polar codes as the channel coding technique for control channels for 5G New radio (NR)communications framework has demonstrated the achievement of Arikan's [1] and will set up their application in practical field. In view of the idea channel polarization, this new coding family is capacity achieving. With preferred or comparable performance over LDPC and turbo codes, it suppresses the performance of the tail-biting convolution codes used in Long Term Evolution (LTE) systems for control channels. This channel coding system is connected for downlink and uplink control information for the enhanced mobile broadband (eMBB) service.

5.1 Introduction to CRC-Aided Decoding

Cyclic redundancy check (CRC) is the most generally utilized system in error detection in the field of data hypothesis and coding. It is also broadly utilized in cellular guidelines. For example, the 3rd generation partnership project (3GPP) includes CRC broadly in the vast majority of its radio access technologies like UMTS, LTE etc. If we consider an output length of N bits of the error detecting encoder and information bits of length A and the length of the CRC sequence as m-bit, i.e. K = A+m. Code rate R would be then characterized as R=K/N. In this chapter we tried to present a combination of both the SC decoder and SCL decoder which is added by CRC that enhance the performance of the polar code. CRC Aided Polar Decoding schematic details are shown in figure 5.1 [28]

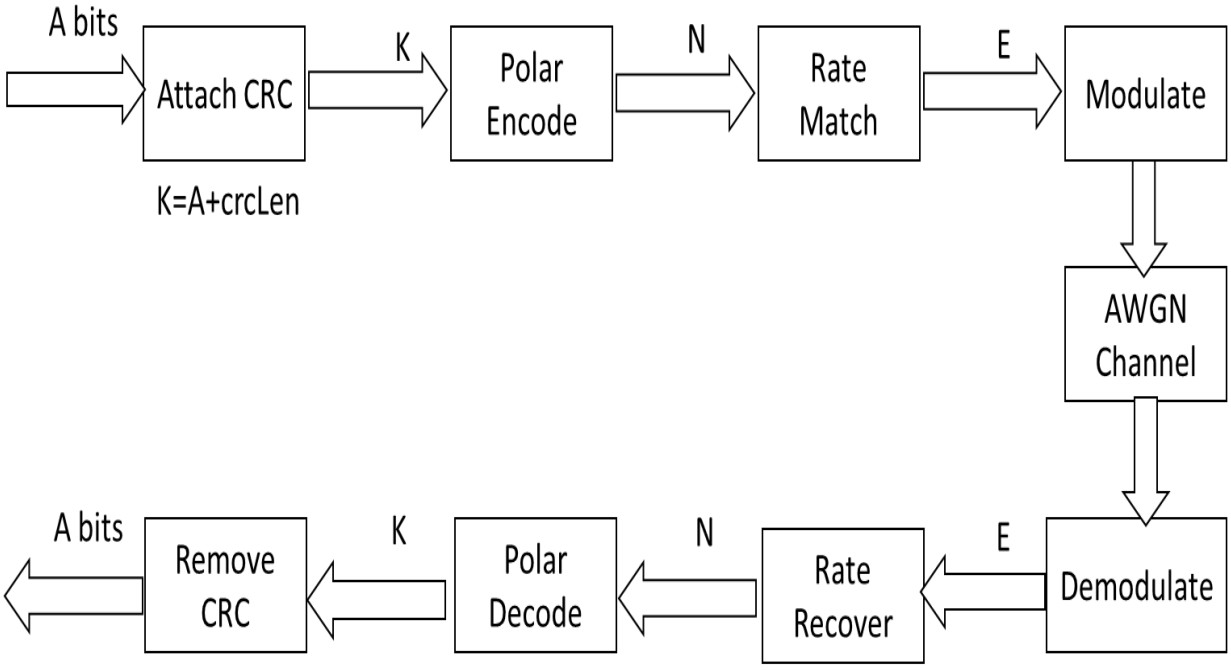


Figure 5.1: Schematic details of CRC aided polar code Decoder

CRC Aided Successive Cancellation Decoding [29] process consists of mainly three steps.

1. Polar Encoding
2. Rate matching and Rate recovery
3. Polar Decoding

5.2 Polar Encoding

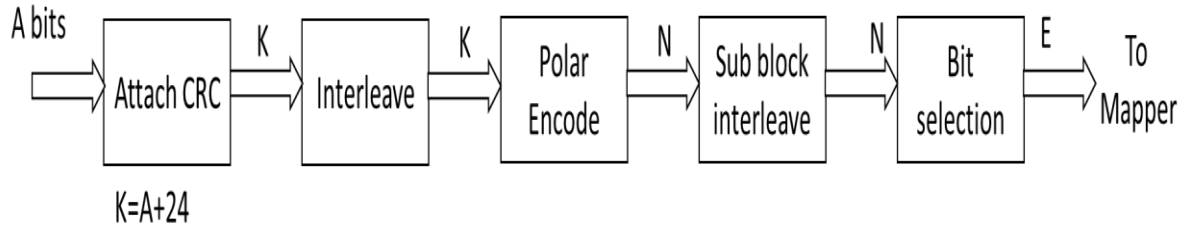


Figure 5.2: CRC Aided polar Encoding [28]

Encoding procedure of a CRC aided SCD is shown here. CRC bits are added at the end of the information bits. After that CRC included bits are then sent for interleaving. Interleaved bits are then encoded. Finally encoder outputs are again interleaved and sent for bit selection.

5.3 Rate Matching and Rate Recovery

The polar encoded sets of bits (N) are rate-coordinated to output the per-defined number of bits (E) for resource component mapping. The coded bits are sub-block interleaved and go to a circular buffer of length N. Depending on the ideal code rate and chose estimations of K, E, and N, either of repetition ($E \geq N$), and puncturing or shortening ($E < N$) is acknowledged by reading the output bits from the buffer.

- For puncturing, E bits are taken from the end
- For shortening, E bits are taken from the start
- For repetition, E bits are repeated modulo N.

For the downlink, the selected bits are passed on to the modulation mapper, while for the uplink; they are further interleaved prior to mapping. At the receiving end, rate recovery is performed for each of the cases-

- For puncturing, corresponding LLRs for the bits removed are set to zero
- For shortening, corresponding LLRs for the bits removed are set to a large value
- For repetition, the set of LLRs corresponding to first N bits are selected.

5.4 Polar Decoding

The implicit CRC encoding of the downlink or uplink message bits dictates the use of the CRC-aided SCL decoding (CA-SCL) [30] as the channel decoder algorithm. It is well known that CRC aided SCL decoding can perform better than turbo or LDPC codes and this was one of the main reason in the selection of polar codes by 3GPP (3^{rd} Generation Partnership Project).

Tal & Vardy [5] describe the SCLD algorithm in terms of likelihood (probabilities). However, due to underflow, the inherent computations are numerically unstable. To overcome this issue, Stimming et.al. [27] proposed the SCL decoding solely in the log-likelihood ratio (LLR) domain. The list decoding is characterized by the L parameter, which represents the number of most likely decoding paths are present. At the end of the successive cancellation decoding, the most reasonable code-path among the L paths is the decoder output. As L is expanded, the decoder execution likewise improves, in any case, with an unavoidable losses impact.

For an input message which is added with CRC aider SCL decoding trim out those paths for which the CRC is invalid, when at least one path has the correct CRC. This additional insight in the final path determination improves the performance further compared to SCL decoding. For the downlink a CRC of 24 bits is used, while for the uplink CRCs of 6 and 11 bits are specified, which vary with the value of k.

Steps that are followed for whole CRC decoding are given bellow step by step.

- K length of crc arbitrary bits are created,
- A CRC is processed and affixed to these bits
- The CRC added bits are polar encoded to the mother code block length
- Rate-matching is performed to transmit E bits
- The E bits are modulated
- White Gaussian Noise of determined power is included
- The noisy signal is soft demodulated to get LLR values
- Rate recovery is done accounting for either of puncturing, shortening or repetition
- The recovered LLR values are polar decoded utilizing the CRC aided SCL decoding technique, including de interleaving.
- Off the decoded K bits, the first K length of crc bits are compared with those transmitted to update the block error and bit error rate.

5.5 CRC-Aided Successive Cancellation List Decoding

The performance of the Successive Cancellation decoder can be improved by using Successive Cancellation List decoding algorithm, which tracks L best decoding paths together. The performance of SCL decoder can be further improved by introducing CRC to SCLD by selecting a valid CRC path among L best decoding paths at the end of decoding. However, SCL decoding algorithm suffers from long latency and low throughput because of high complexity, O(L N log N) calculations as L and N increases. The throughput of the SCLD can be improved by using an adaptive decoder [31], which provides the SC decoding throughput with the SCL performance.

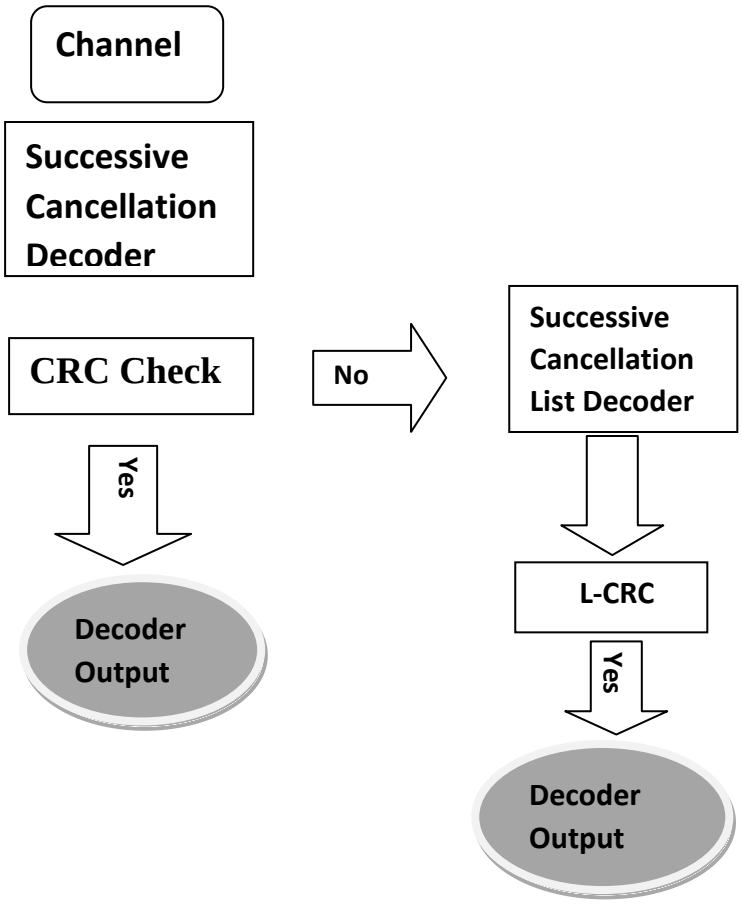


Figure 5.3: Flow chart of the CRC aided List decoder [5]

The decoder has mainly three sections, SC decoder, SCL decoder and CRC section. Initially, the SC decoder is activated and a decision evaluation vector is calculated. After that, the CRC decoder controls whether the decision evaluation vector is correct. If the CRC is valid, the decision evaluation vector is quite likely to be the correct information vector. In this case the

adaptive decoder is immediately shut down without the activation of the SCL decoder. In other case, when the CRC is invalid, the SCL decoder is activated and L information vector candidates are generated. Among these vectors, the CRC decoder selects a candidate, which has a valid CRC vector. If more than one CRC vector candidate is valid, the most probable one among these candidates is selected. At the end, when none of the candidate has a valid CRC vector, the CRC decoder chooses the most probable decision estimation vector to reduce BER.

Chapter 6

FPGA Implementation of Successive Cancellation Decoder

Since late 1970's Small Scale Integrated Circuits (SSIs) started replacing digital hardware realized with discrete components. Early 1980's has seen a significant progress in the design of Very Large Scale Integrated Circuits (VLSI). Large demand of such processors enforced the semiconductor industries to produce those in large volume. Full custom ICs, which are fully designed and implemented on VLSI by the manufacturer themselves, however, are not appropriate for specialized or dedicated applications as the applications do not require bulk quantities of such ICs. Application Specific Integrated Circuits (ASIC) are the silicon chips that are designed for specific applications. They cannot be used for general purposes. They are full custom ICs. User has no freedom to change the architecture. As they are built for a particular task, it provides the highest speed for implementation. Semi-custom ICs are those where users, according to their need, can program the connectivity between the components of the ICs. Thus, it evolved in late 1990's to bridge the gap between full custom ICs and discrete realization of circuits [34]. Programmable Logic Devices (PLDs) and Gate arrays which evolved in the era of new millennium fall in the category of semi-custom ICs. Gate arrays include an assembly of transistors or logic gates on a two dimensional array in the custom phase Programmable Logic Devices (PLDs) and Gate arrays which evolved in the era of new millennium fall in the category of semi-custom ICs. Gate arrays include an assembly of transistors or logic gates on a two dimensional array in the custom phase production of the ICs.

6.1 FPGA

Field programmable Gate Array (FPGA) [32] is one of the special types of gate array devices, which when shipped to the user, is shipped with general purpose pre-fabricated metallization, often with variable length segment and routing tracks. The device is programmed by turning on switches to establish connectivity between the circuit nodes and metal routing tracks. The connections are made by transistors switches, which are controlled by a programming memory element or by anti fuses. FPGAs thus classically differ from the full custom VLSI from the point of views of their flexibility of programming in the fields according to user's convenience. Its disadvantage is that in general, the power consumption of FPGA is more compared to ASIC.

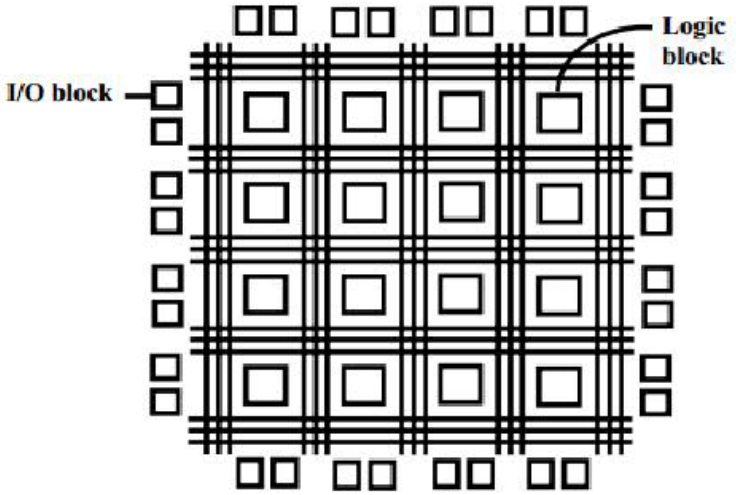


Figure 6.1: Internal Structure of FPGA

6.2 Problem Formulation

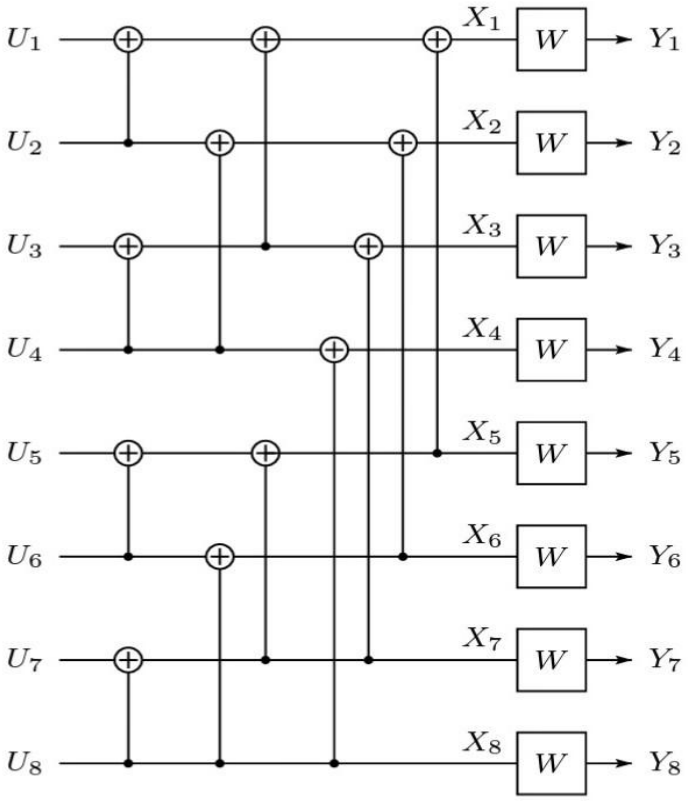


Figure 6.2: The factor graph representation of 8-bit Decoder

Figure 6.2 shows a basic graphical representation of an 8bit SC Decoder. From the graph it is clearly shown that the number of XOR gates used in the Decoding is 12, but when we are constructing the decoding equation number of XOR operation becoming 19 which is shown

$Y_8 = u_8$

$Y_7 = u_7 \oplus u_8$

$Y_6 = u_6 \oplus u_8$

$Y_5 = u_8 \oplus u_7 \oplus u_6 \oplus u_5$

$Y_4 = u_8 \oplus u_4$

$Y_3 = u_8 \oplus u_7 \oplus u_4 \oplus u_3$

$Y_2 = u_8 \oplus u_6 \oplus u_4 \oplus u_2$

$Y_1 = u_1 \oplus u_2 \oplus u_3 \oplus u_4 \oplus u_5 \oplus u_6 \oplus u_7 \oplus u_8$

If we use common sub expression elimination then the numbers of XOR operations can be reduced. The compact expressions are as follows.

$Y_8 = u_8$

$Y_7 = \mathbf{(u_8 \oplus u_7)}$

$Y_6 = \mathbf{(u_8 \oplus u_6)}$

$Y_5 = \{(u_8 \oplus u_7) \oplus (u_6 \oplus u_5)\} = \mathbf{(Y_7 \oplus c_1)}$ [let $c_1 = \mathbf{u_6 \oplus u_5}$]

$Y_4 = \mathbf{(u_8 \oplus u_4)}$

$Y_3 = \{(u_8 \oplus u_7) \oplus (u_4 \oplus u_3)\} = \mathbf{(Y_7 \oplus c_2)}$ [let $c_2 = (\mathbf{u_4 \oplus u_3})$]

$Y_2 = \{(u_8 \oplus u_6) \oplus (u_4 \oplus u_2)\} = (\mathbf{Y_6 \oplus c_3})$ [let $c_3 = (\mathbf{u_4 \oplus u_2})$]

$Y_1 = [\{(u_1 \oplus u_2) \oplus (u_3 \oplus u_4)\} \oplus \{(u_5 \oplus u_6) \oplus (u_7 \oplus u_8)\}] = \{(c_2 \oplus c_4) \oplus (c_1 \oplus Y_7)\}$

[let $c_4 = (\mathbf{u_1 \oplus u_2})$]

$= (\mathbf{c_5 \oplus Y_5})$ [let $c_5 = (\mathbf{c_4 \oplus c_2})$]

By using common sub expressions elimination, the number of XOR operations is reduced from19 to 12.A table is prepared which gives an idea how much performance improvement can be achieved by using common sub expression elimination technique.

Table 6.1 Performance Improvement using common sub expression

Code Length (N)	No of XOR operation used (without comm. Sub exp. elimination)	No of XOR operation used (with comm. Sub exp. elimination)	Performance Improvement (%)
8	19	12	36.84
16	65	32	50.76
32	211	80	62.08
64	661	192	70.95

This is the main object of this section. We have used this method for compact design of decoder to reduce latency and memory space required. We have used four short length codes (N=8, 16, 32, 64) for this implementation.

We have used Xilinx ISE Design suite 14.7 software for the implementation. Verilog Hardware Description Language is used to express each design. In this work, during synthesis FPGA based virtex 6 (xc6vlx75t1-1 L-ff484) target device is used.

6.3 FPGA-based Implementation Result

6.3.1 For code length N=8

Figure 6.3: Input to Decoder (N=8)

Figure 6.4: Simulated output (N=8)

Table 6.2: Design summery (N=8)

Slice Logic Utilization	Used	Available	Utilization
No. of Slice LUTs	5	46,560	1%
No. of used Logic	5	46,560	1%
No. of occupied Slice	3	11,640	1%
No. of LUT Flip Flop pair used	5	5	100%
No. of Bonded (IOBs)	16	240	7%
No. of Startup	1	1	100%

Delay	1.48 ns
Memory used	281916 kilobytes

6.3.2 For code length N=16

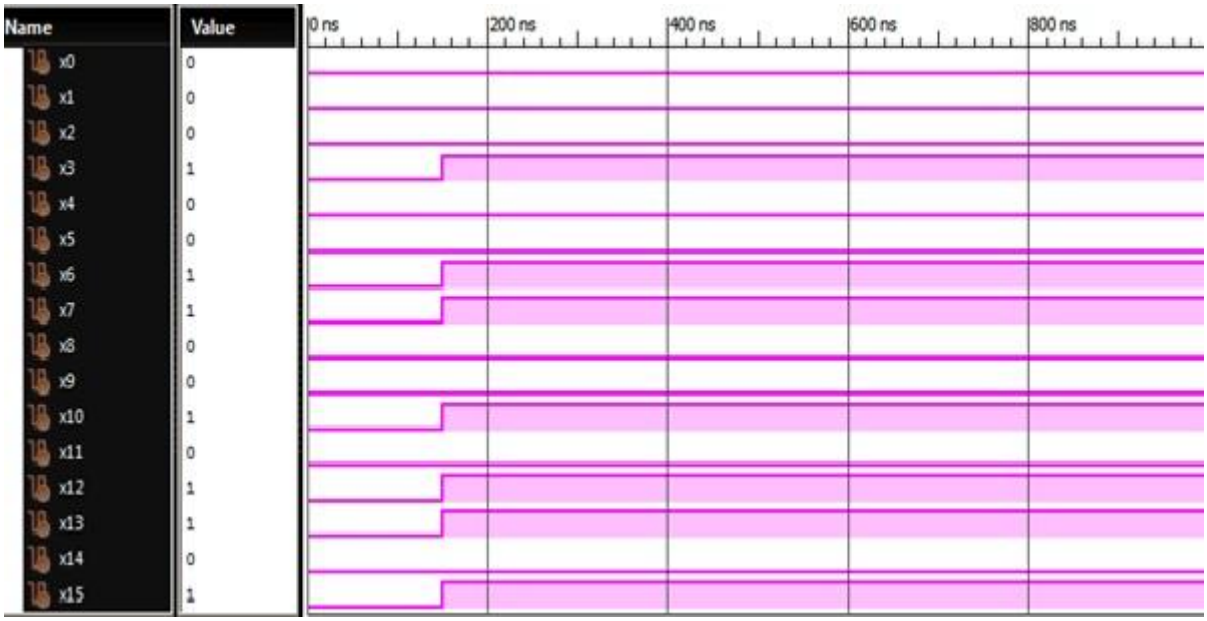


Figure 6.5: Input to Decoder (N=16)

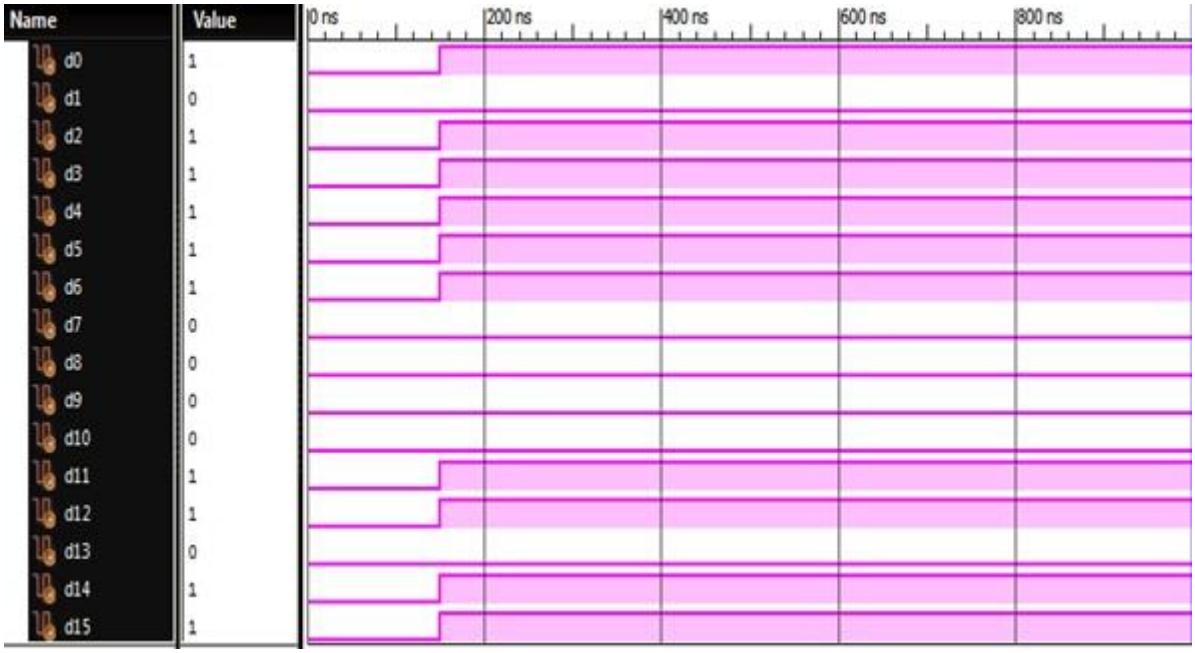


Figure 6.6: Simulated output (N=16)

Table 6.3: Design summery (N=16)

Slice Logic Utilization	Used	Available	Utilization
No. of Slice LUTs	14	46,560	1%
No. of used Logic	14	46,560	1%
No. of occupied Slice	9	11,640	1%
No. of LUT Flip Flop pair used	14	14	100%
No. of Bonded (IOBs)	32	240	13%
No. of Start up	1	1	100%

Delay	1.612 ns
Memory used	281916 kilobytes

6.3.3 For code length N=32

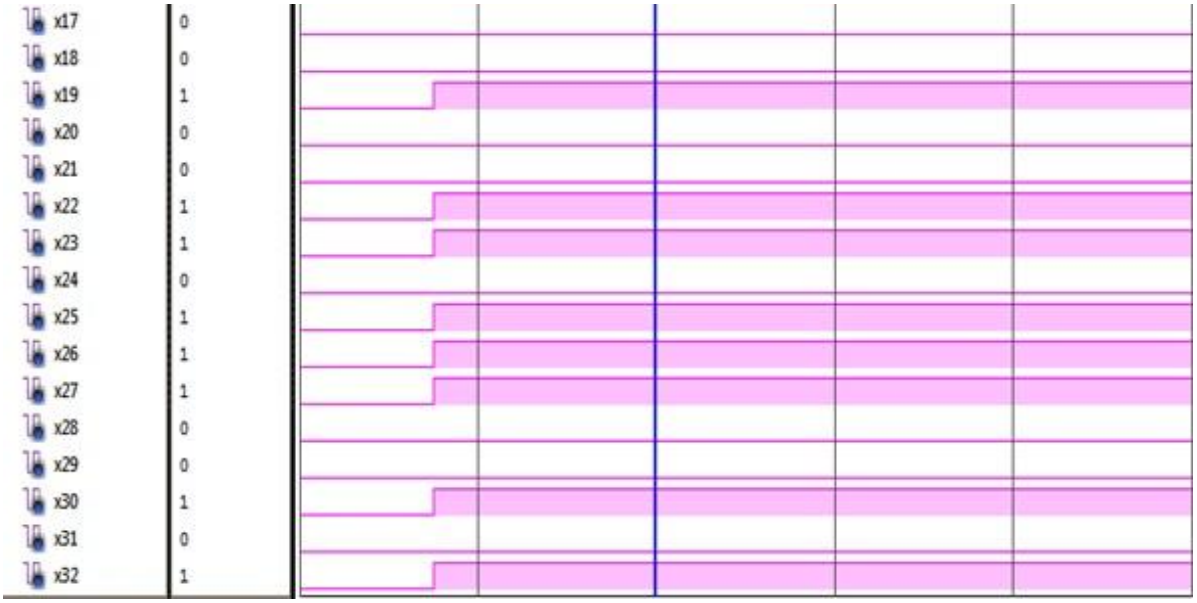


Figure 6.7: Input to Decoder (N=32)

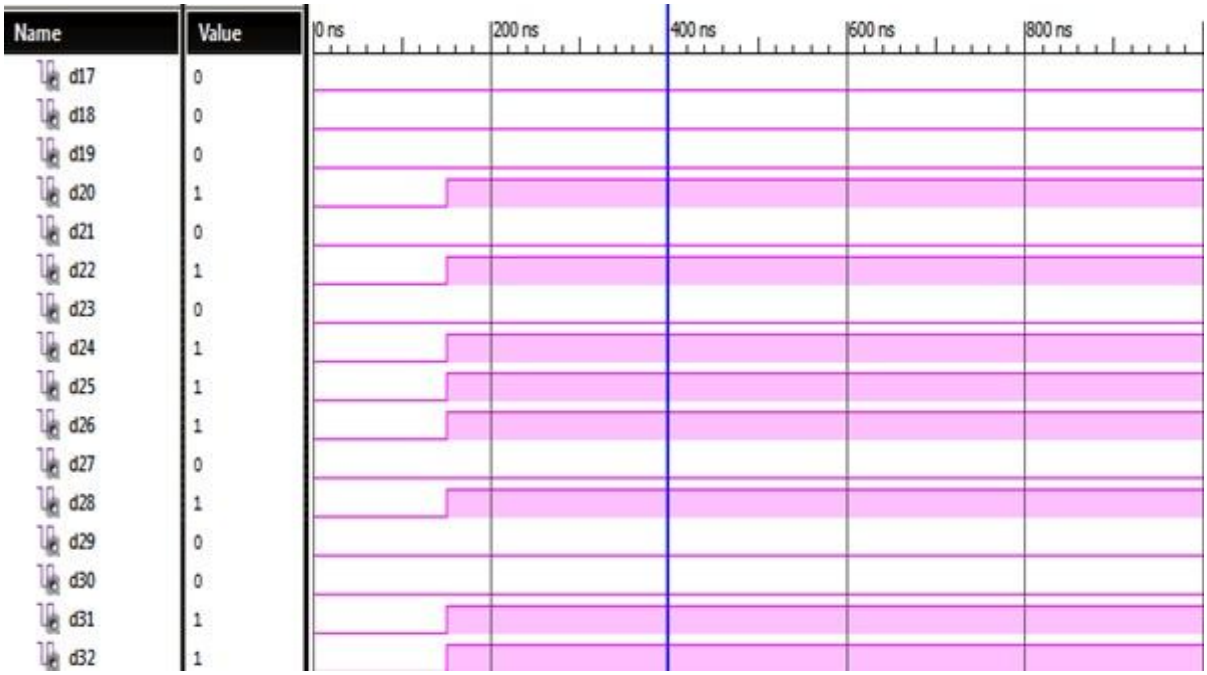


Figure 6.8: Simulated output (N=32)

Table 6.4: Design summery (N=32)

Slice Logic Utilization	Used	Available	Utilization
No. of Slice LUTs	44	46,560	1%
No. of used Logic	44	46,560	1%
No. of occupied Slice	26	11,640	1%
No. of LUT Flip Flop pair used	44	44	100%
No. of Bonded (IOBs)	64	240	27%
No. of Start up	1	1	100%

Delay	1.97 ns
Memory used	281916 kilobytes

6.3.4 For code length N=64

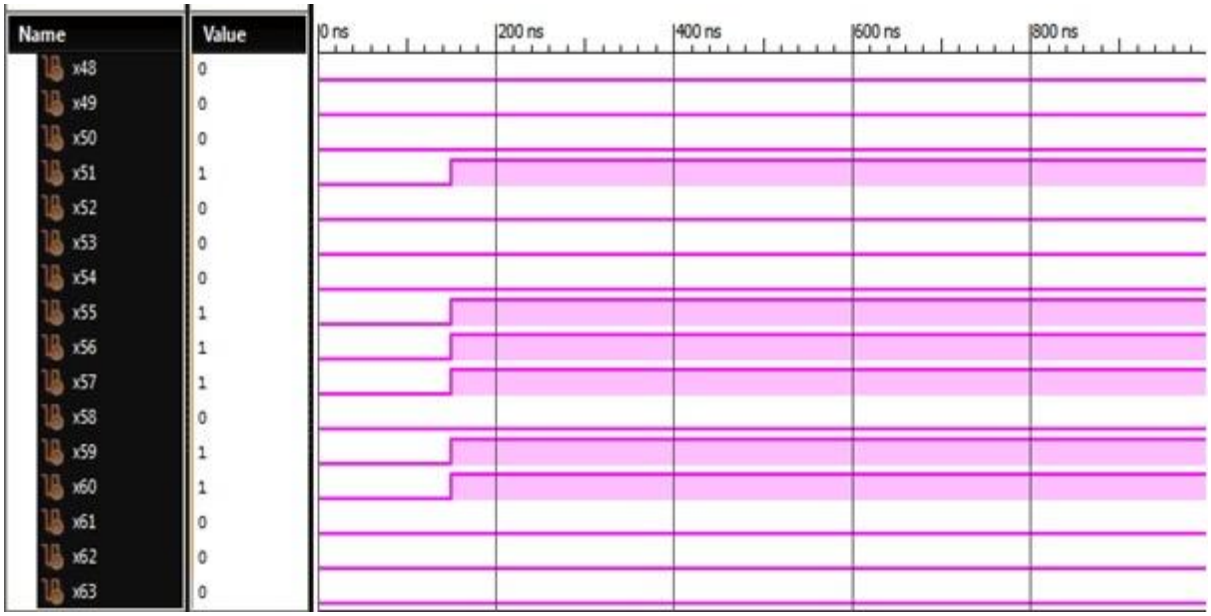


Figure 6.9: Input to Decoder (N=64)

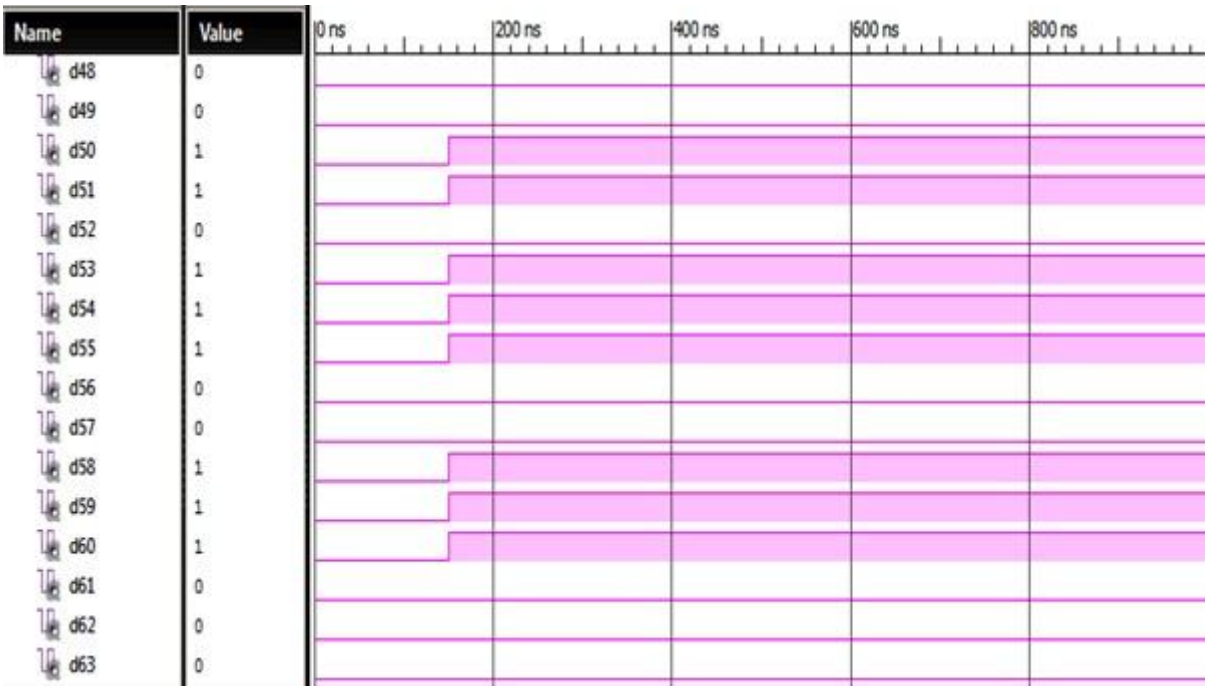


Figure 6.10: Simulated output (N=64)

Table 6.5: Design summery (N=64)

Slice Logic Utilization	Used	Available	Utilization
No. of Slice LUTs	80	46,560	1%
No. of used Logic	80	46,560	1%
No. of occupied Slice	46	16,720	1%
No. of LUT Flip Flop pair used	80	-	-
No. of Bonded (IOBs)	128	240	53%
No. of Start up	1	1	100%

Delay	3.547 ns
Memory used	282940kilobytes

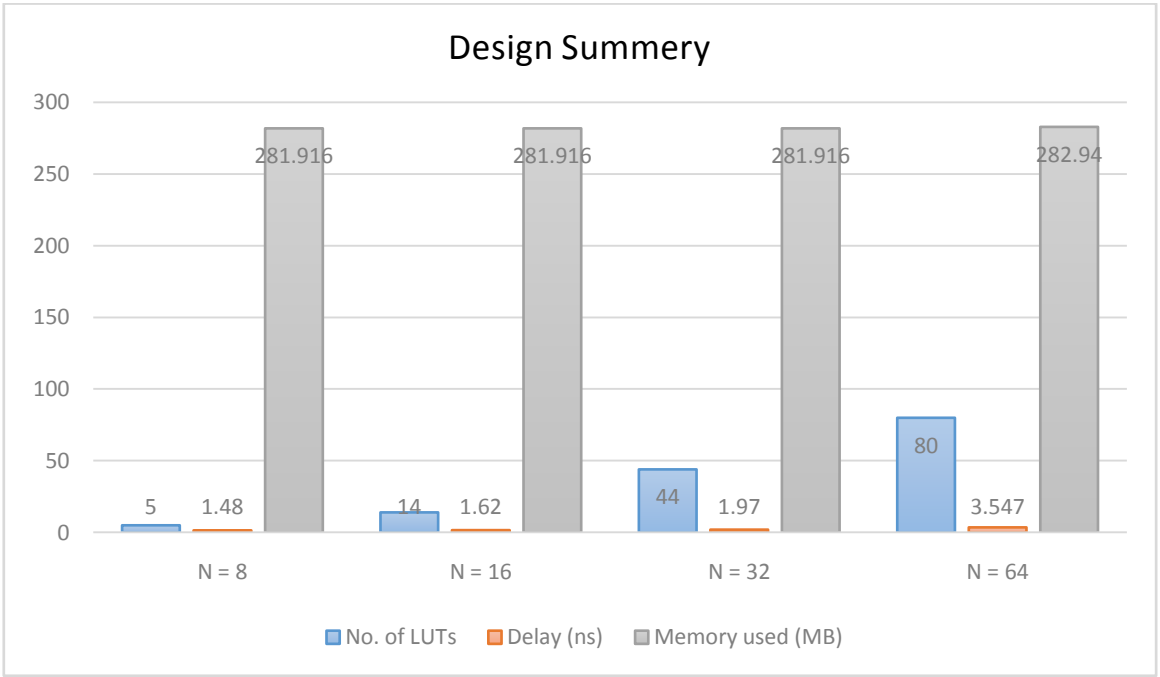


Figure 6.11: Bar chart representation of Design Summery

6.4 Summary

Implementation of FPGA-based successive cancellation decoder is shown in this chapter. We have used short length (N=8, 16, 32, 64) code word for the implementation. Verilog Hardware Description Language is use for the implementation of each module. We have tried to make the design compact by reducing the numbers of XOR operations compared to traditional implementation. By reducing the numbers of XOR operations, we have achieved up to 70% improvement in area overhead. As the length of code word increase latency of computation also increase

Chapter 7

Performance of Polar Codes

In this chapter performance of polar codes based on SCD and CRC-aided SCD has been analysed. In this analysis, the performance of polar codes has been shown for short codewords length. Short length codes are mainly used in practical situations such as wireless communication but their performance is not very impressive. Therefore, the enhancement of polar codes performance is an important issue.

7.1 Successive Cancellation Decoding

In this work we have mainly used short block length code word (up to N=2048) over BEC channel. Our analysis mainly focus on performance in terms of Bit Error Rate (BER) vs. Eb/No (dB) and Frame Error Rate (FER) vs. Eb/No (dB) mainly for different code rate, different code length.

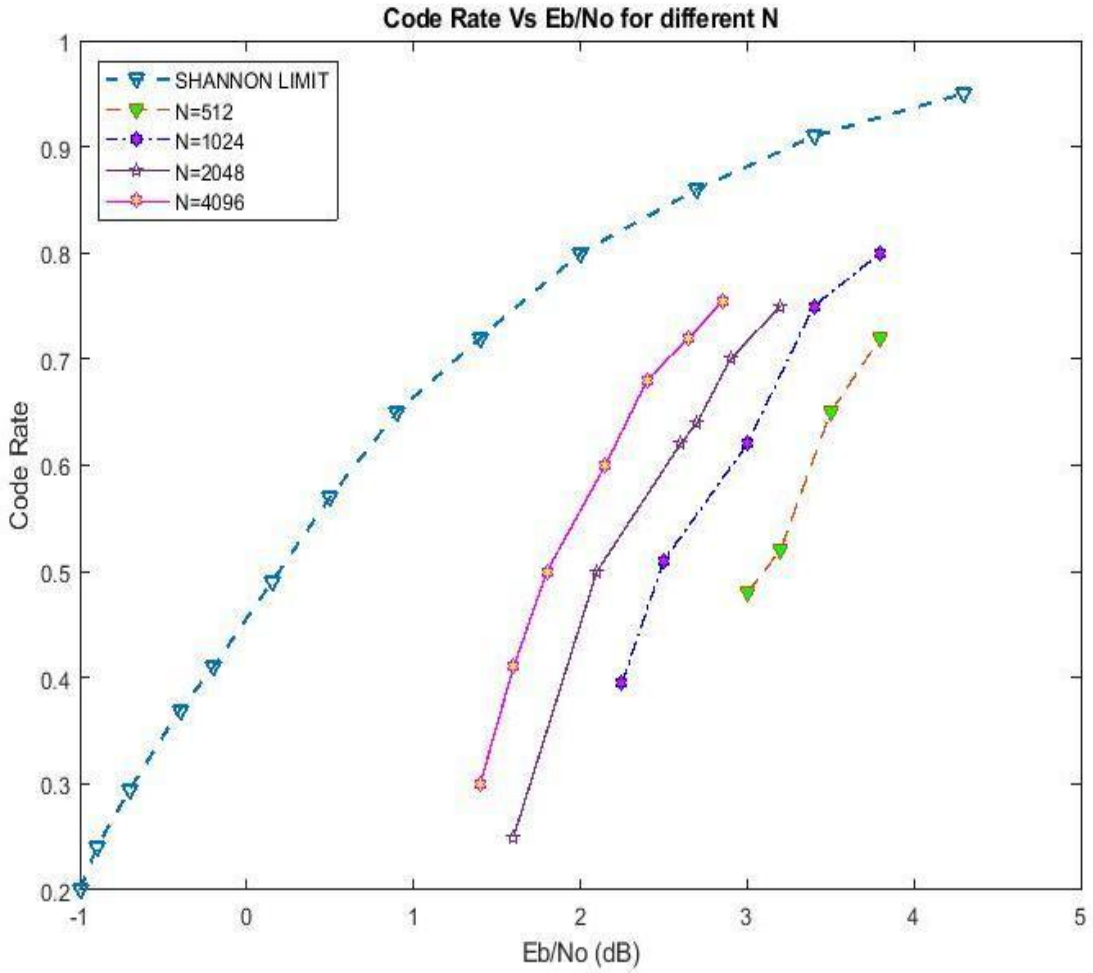


Figure 7.1: Code rate vs. Eb/No (dB) for different code length

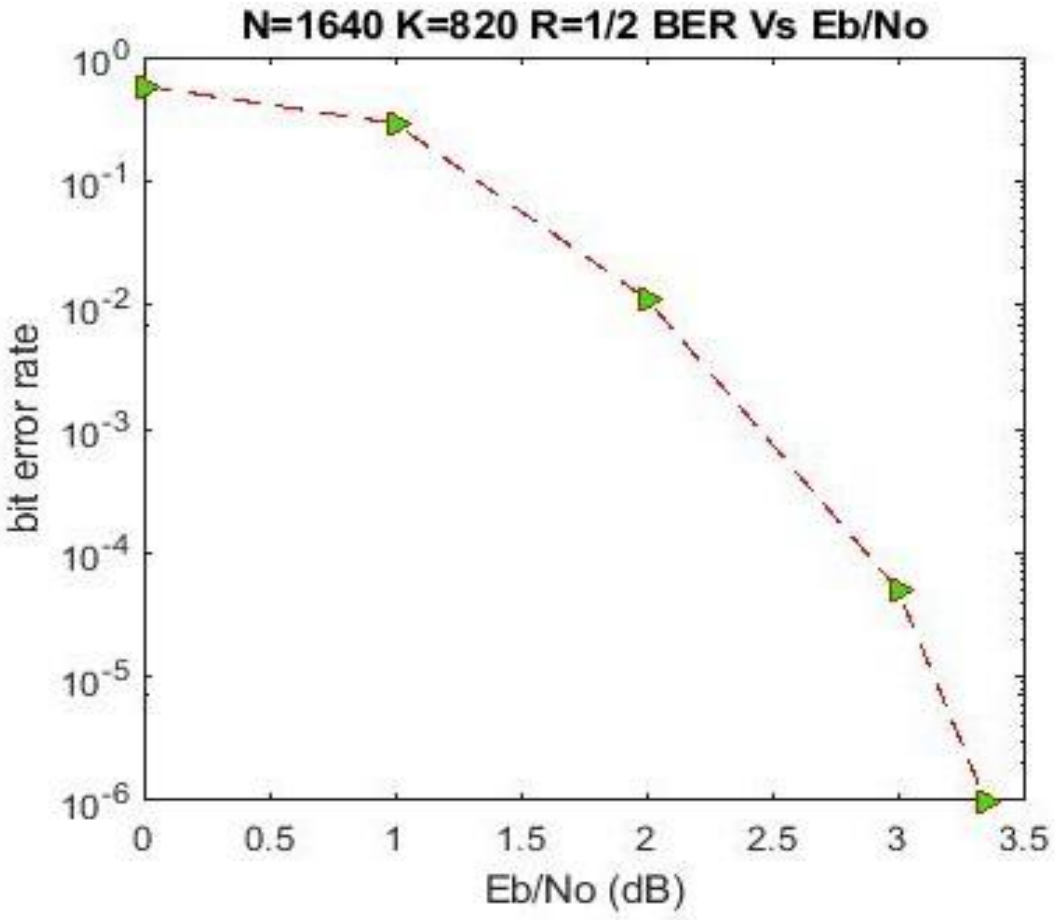


Figure 7.2: Bit error rate vs. Eb/No (dB) for N=1640, K=820, R=1/2

Table 7.1: Bit Error Rate vs. Eb/No (dB) (N=1640, K=820, R=1/2)

Eb/No (dB)	Bit Error Rate
0	0.5832139
1	0.2915001
2	0.0112148
3	0.0000513
3.35	0.0000011

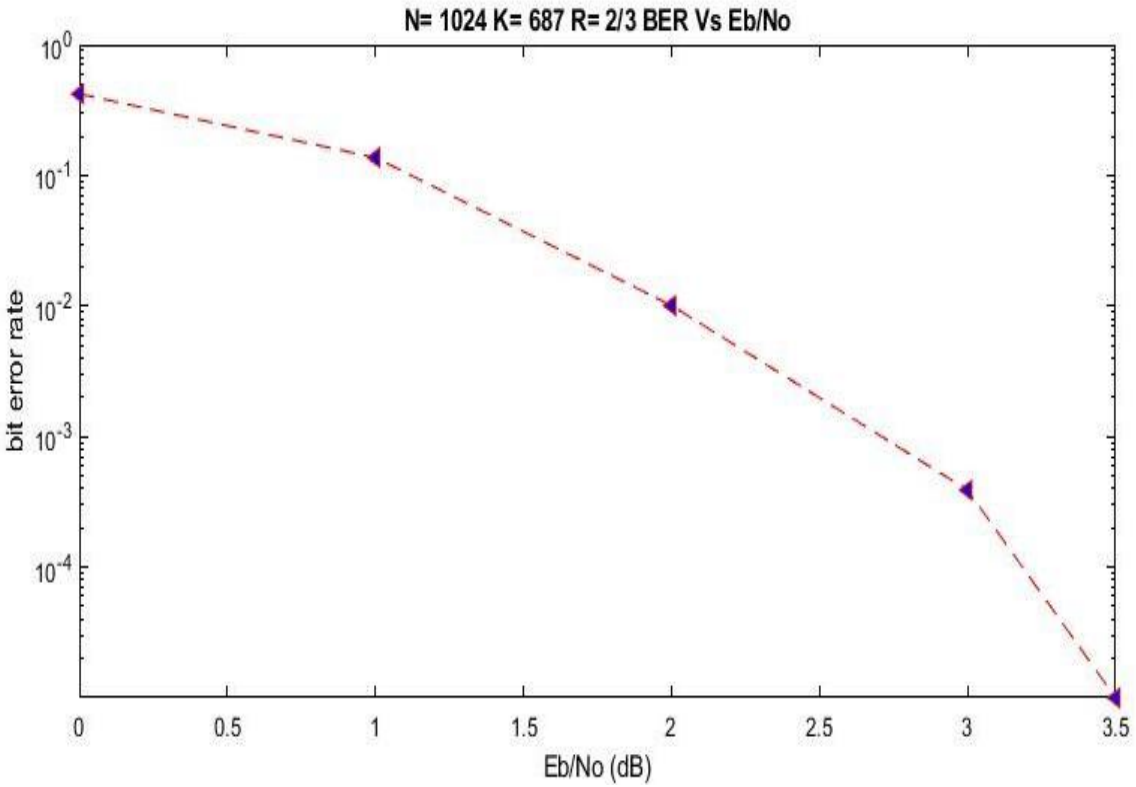


Figure 7.3: Bit error rate vs. Eb/No (dB) for N=1024, K=687, R=2/3

Table 7.2: Bit error rate vs. Eb/No (dB) (N=1024, K=687, R=2/3)

Eb/No (dB)	Bit Error Rate
0	0.4742103
1	0.2313710
2	0.0101321
3	0.0003917
3.5	0.0000101

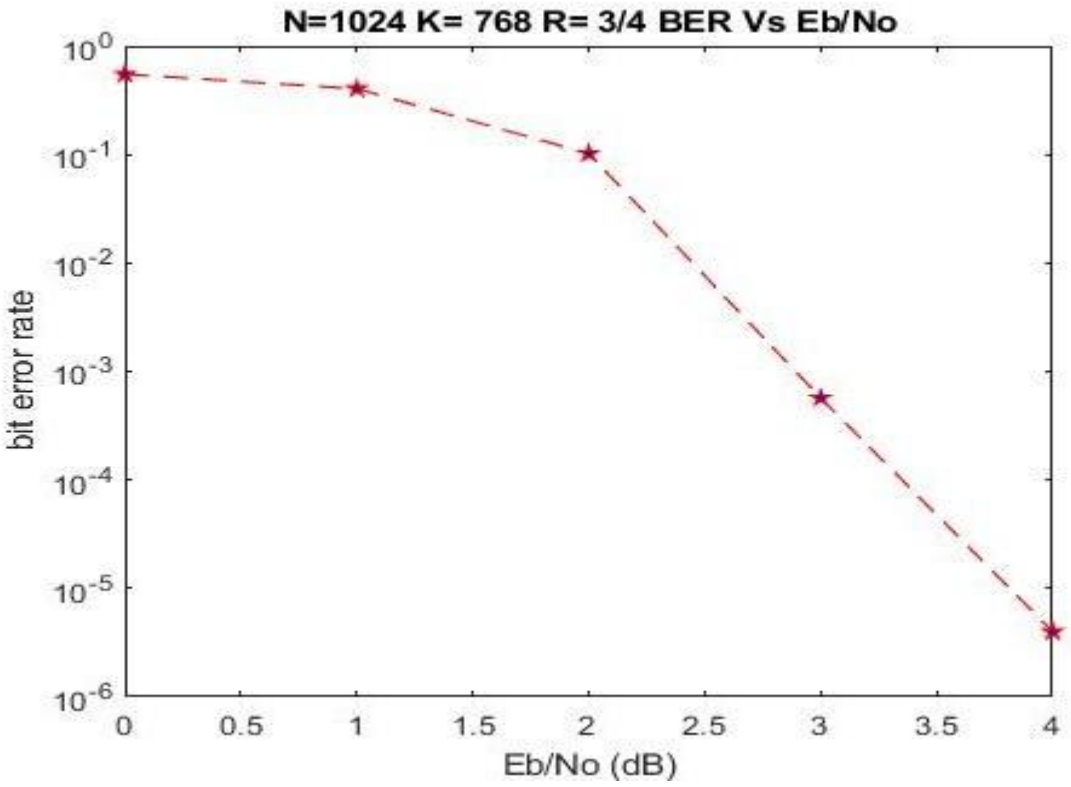


Figure 7.4: Bit error rate vs. Eb/No (dB) for N=1024, K=768, R=3/4

Table 7.3: Bit error rate vs. Eb/No (dB) (N=1024, K=768, R=3/4)

Eb/No (dB)	Bit Error Rate
0	0.5255041
1	0.4053173
2	0.1021309
3	0.0005713
4	0.0000042

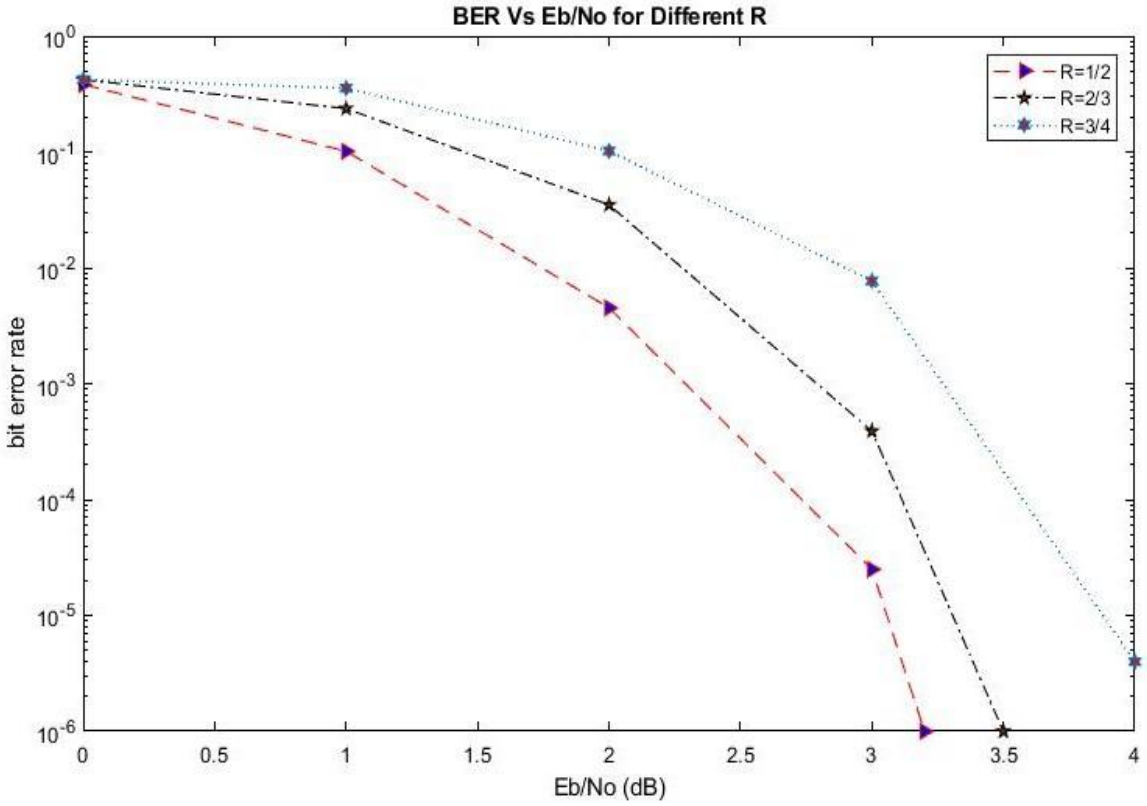


Figure 7.5: Bit error rate vs. Eb/No (dB) for different code rates

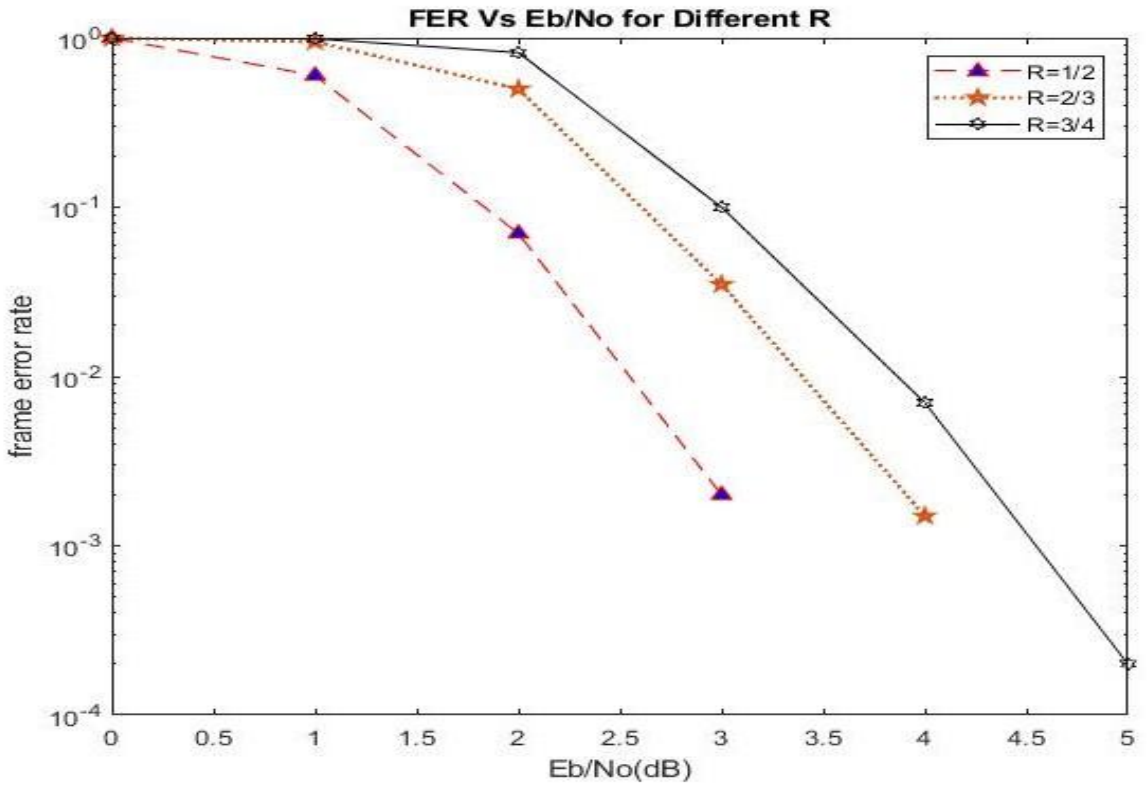


Figure 7.6: Frame error rate vs. Eb/No (dB) for different code rates

Table 7.4: Frame error rate vs. Eb/No (dB) (different code rates)

Eb/No (dB)	Frame Error Rate		
	R=1/2	R=2/3	R=3/4
0	1.0	1.0	1.0
1	0.65321	0.90734	0.98010
2	0.08164	0.70129	0.85231
3	0.00437	0.07716	0.12437
4	0.00010	0.00132	0.00261
5	0.00000	0.00010	0.00022

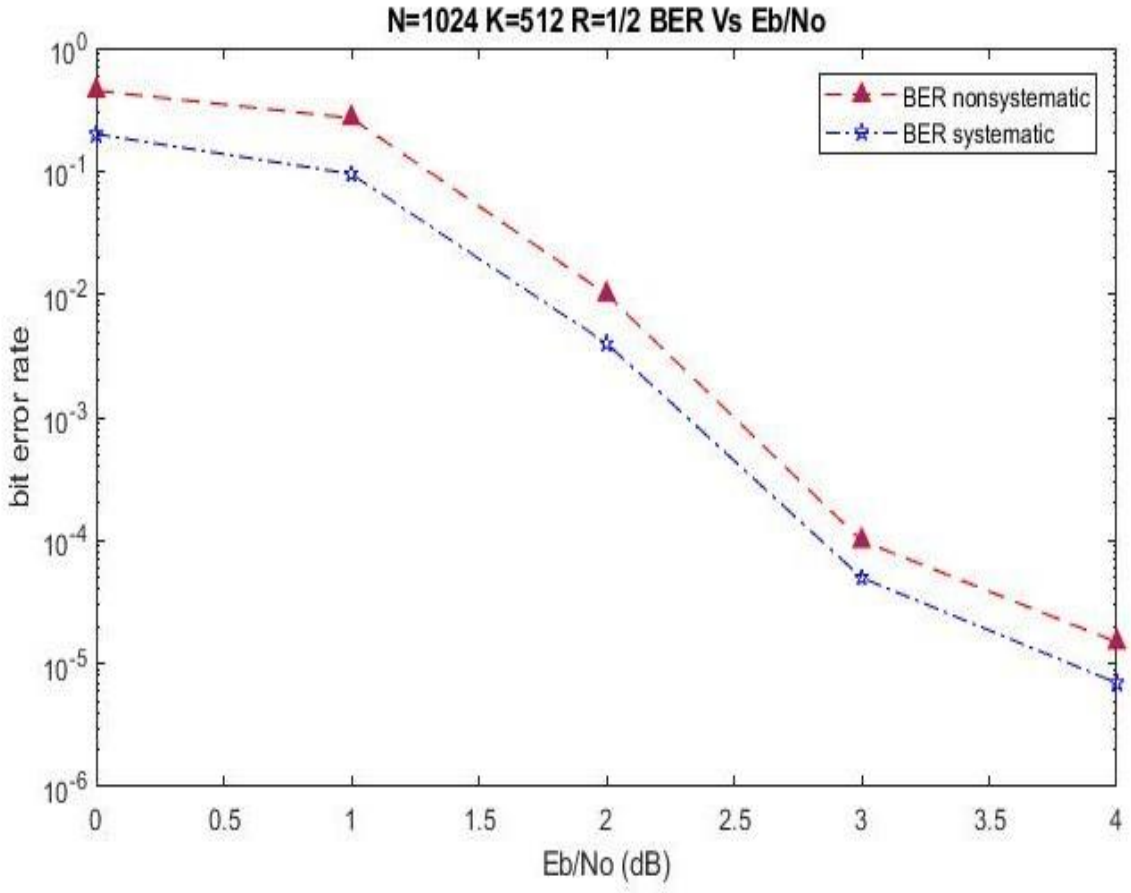


Figure 7.7: Bit error rate vs. Eb/No (dB) for non-systematic and systematic code with code rate R=1/2

Table 7.5: Bit error rate vs. Eb/No (dB) (systematic and non-systematic code, R=1/2)

Eb/No (dB)	Bit Error Rate	
	Systematic	Non-systematic
0	0.2001372	0.4527130
1	0.0957321	0.2752314
2	0.0040237	0.0112475
3	0.0000526	0.0001519
4	0.0000071	0.0000154

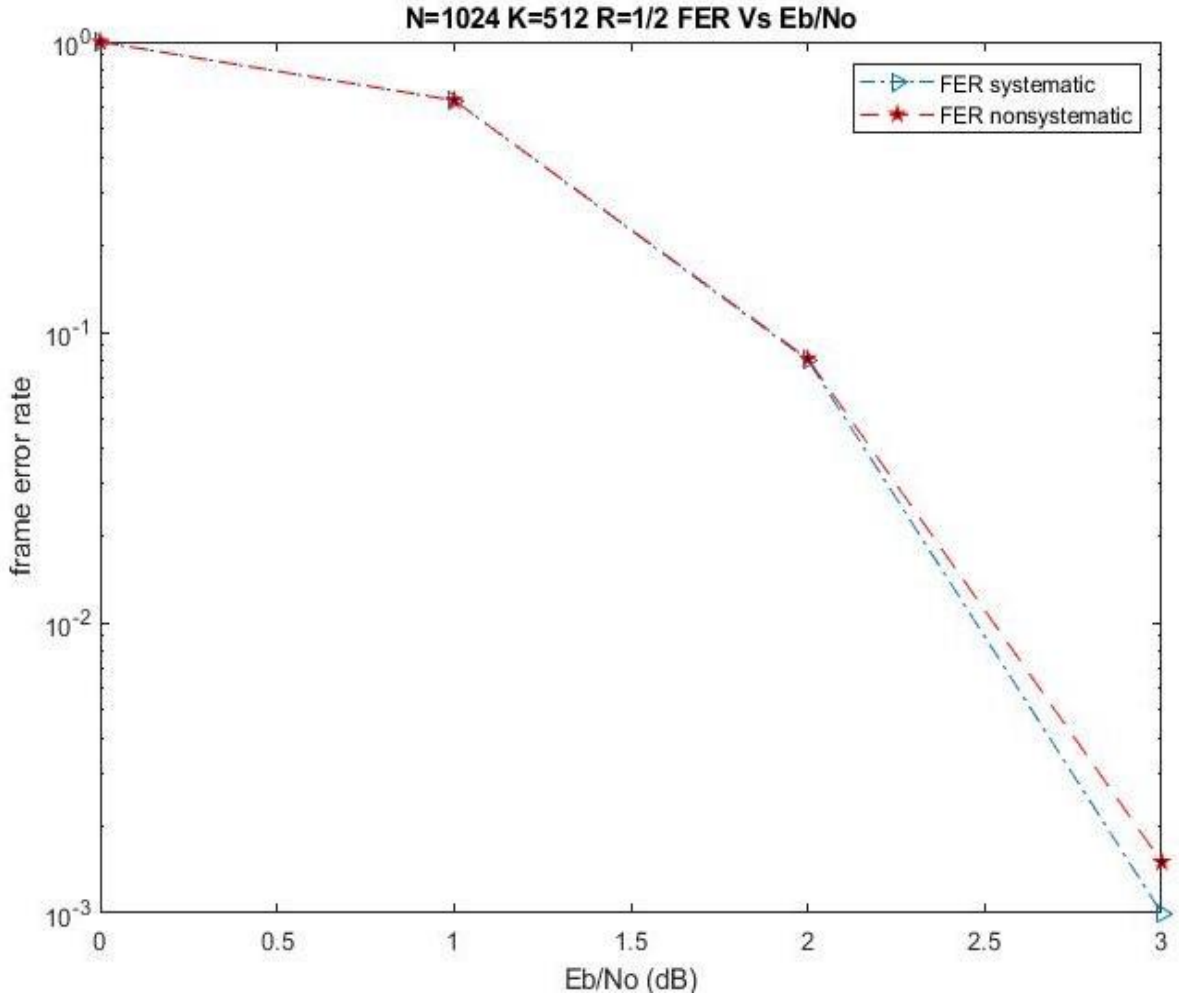


Figure 7.8: Frame error rate vs. Eb/No (dB) for non-systematic and systematic code with code rate R=1/2

Table 7.6: Frame error rate vs. Eb/No (dB) (systematic and non-systematic code, R=1/2)

Eb/No (dB)	Frame Error Rate	
	Systematic	Non-systematic
0	0.99903	1.0
1	0.62903	0.62923
2	0.08017	0.08116
3	0.00102	0.00158

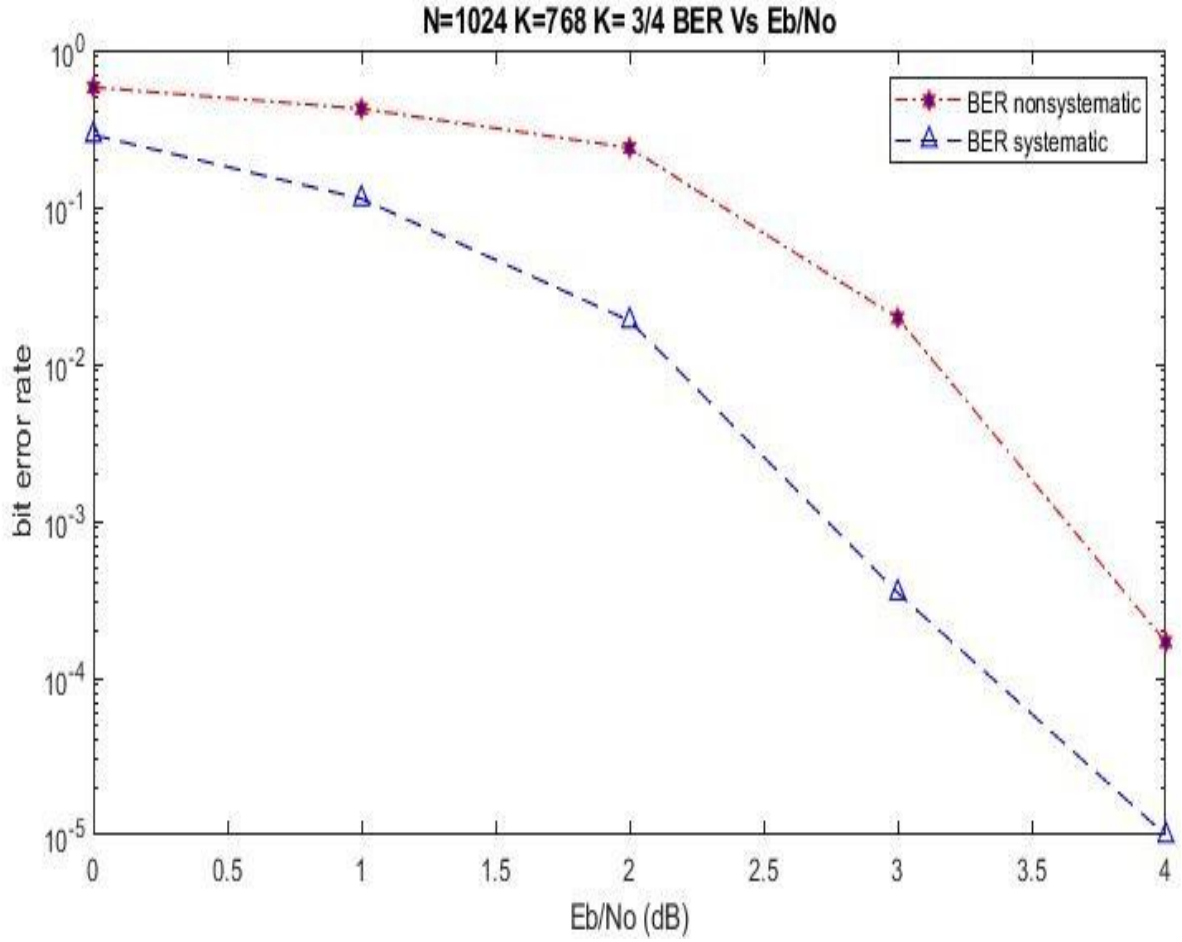


Figure 7.9: Bit error rate vs. Eb/No (dB) for non-systematic and systematic polar code with code rate R=3/4

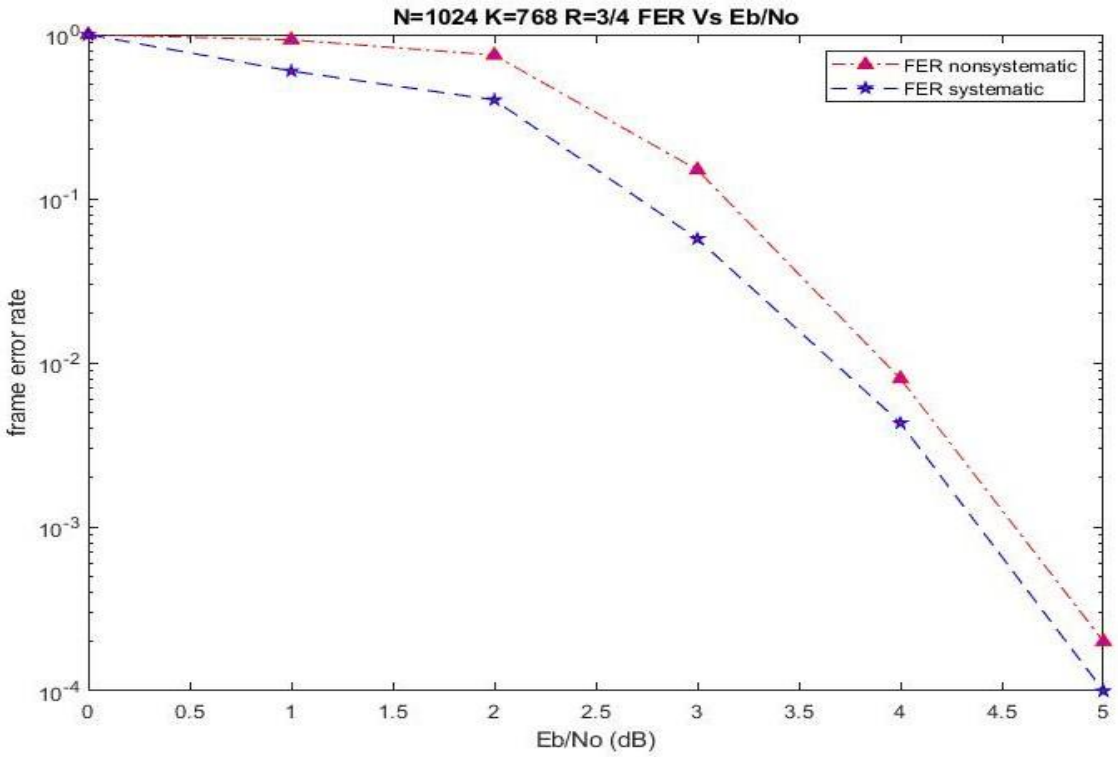


Figure 7.10: Frame error rate vs. Eb/No (dB) for non-systematic and systematic polar code with code rate R=3/4

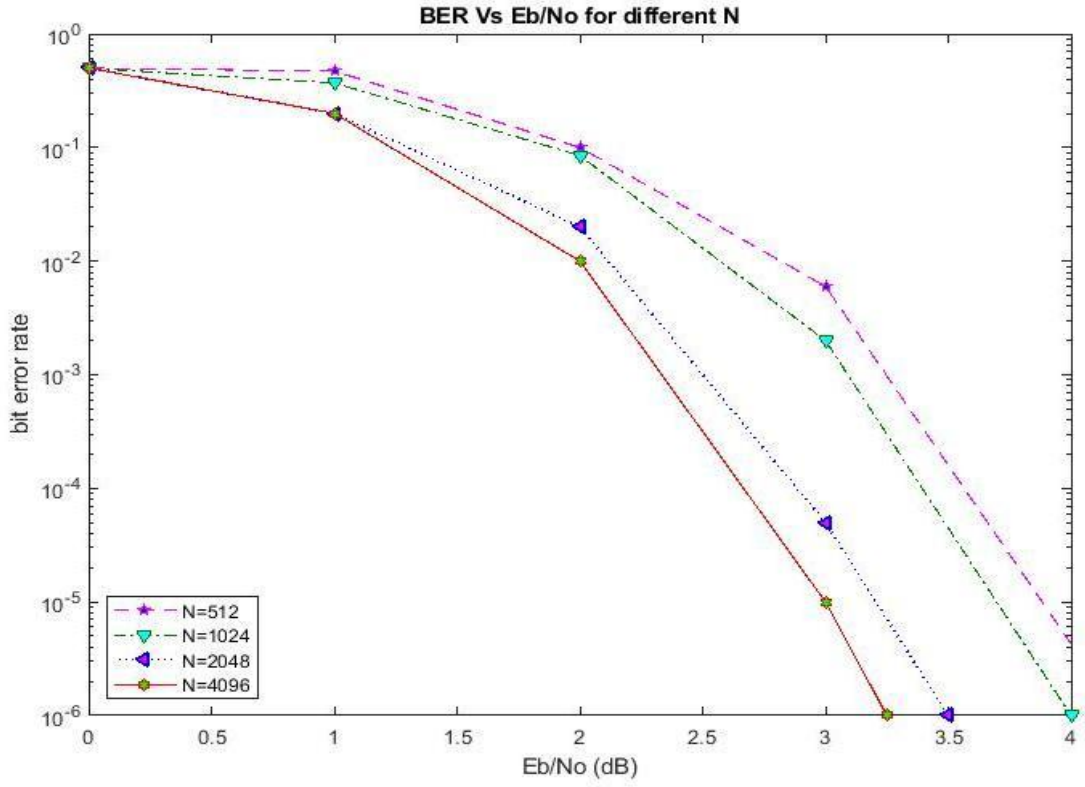


Figure 7.11: Bit error rate vs. Eb/No (dB) for different code length

Table 7.7: Bit error rate vs. Eb/No (dB) (different values of N)

Eb/No (dB)	Bit Error Rate			
	N=512	N=1024	N=2048	N=4096
0	0.5324713	0.5271341	0.5172097	0.5092173
1	0.4731235	0.3735107	0.2713790	0.2372917
2	0.1021879	0.0857329	0.0267921	0.0197152
3	0.0068197	0.0029316	0.0000524	0.0000101
4	0.0000072	0.0000010	0.0000000	0.0000000

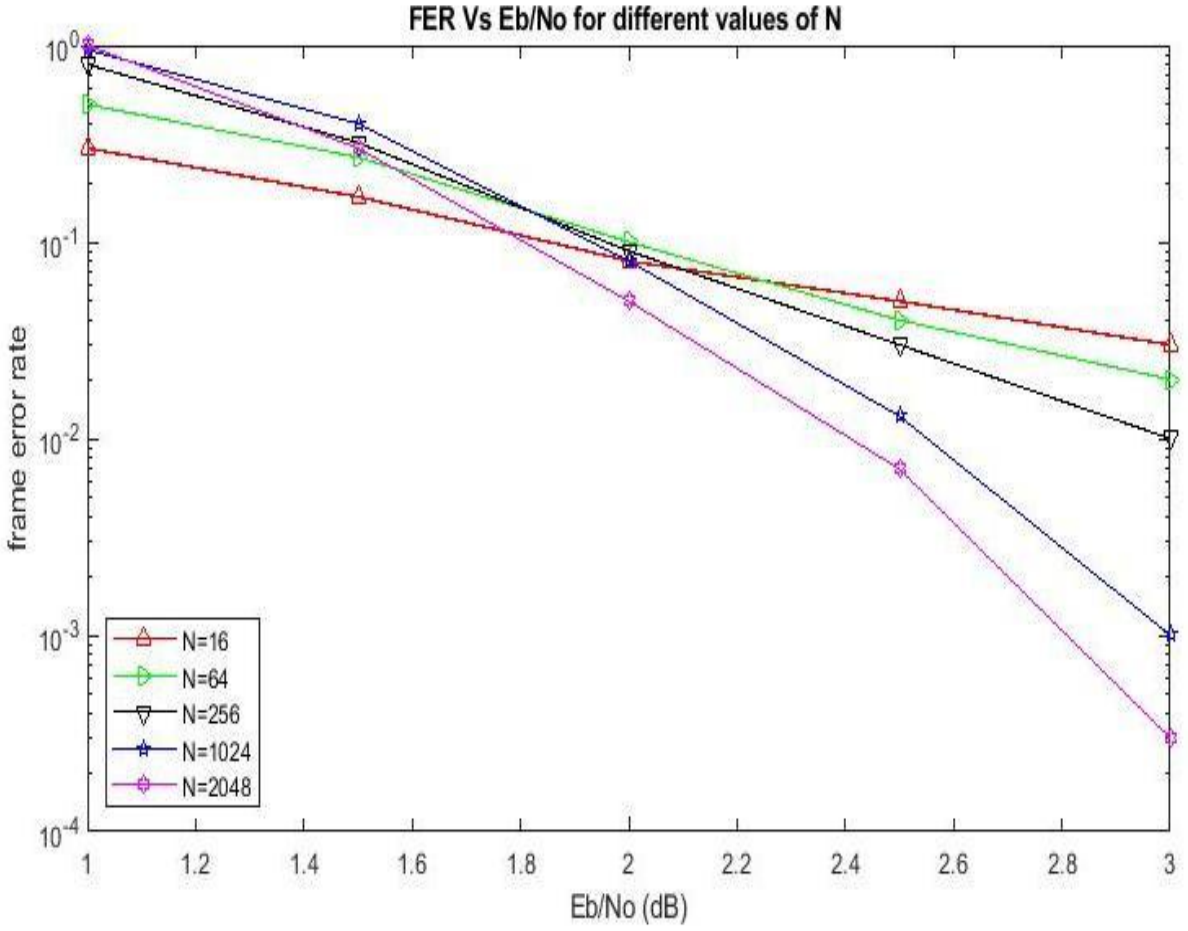


Figure 7.12: Frame error rate vs. Eb/No (dB) for different code length

Table 7.8: Frame error rate vs. Eb/No (dB) (different values of N)

Eb/No (dB)	Frame Error Rate				
	N=16	N=64	N=256	N=1024	N=2048
1	0.31275	0.51844	0.88724	0.95657	0.1
1.5	0.17934	0.27447	0.63981	0.74532	0.38698
2	0.08547	0.10975	0.09753	0.08127	0.51742
2.5	0.05197	0.04824	0.03951	0.01391	0.00741
3	0.03718	0.02917	0.01852	0.00137	0.00037

7.2 CRC-Aided Successive Cancellation Decoding

Performance analysis of SC decoding is challenging due to correlation between codeword bits. For a binary erasure channel (BEC), Parizi and Telatar have shown that the correlations between the erasure events decay fast and thus the union bound on the frame error probability becomes tight as the codeword length increases [1]

In this section we have tried to find performance analysis of SCD with some fixed CRC and list size. We have tried to find performance of CRC aided SCD in terms of block error rate vs. SNR (dB) and bit error rate vs. SNR (dB) for different code rates, different list sizes and finally for different modulation schemes.

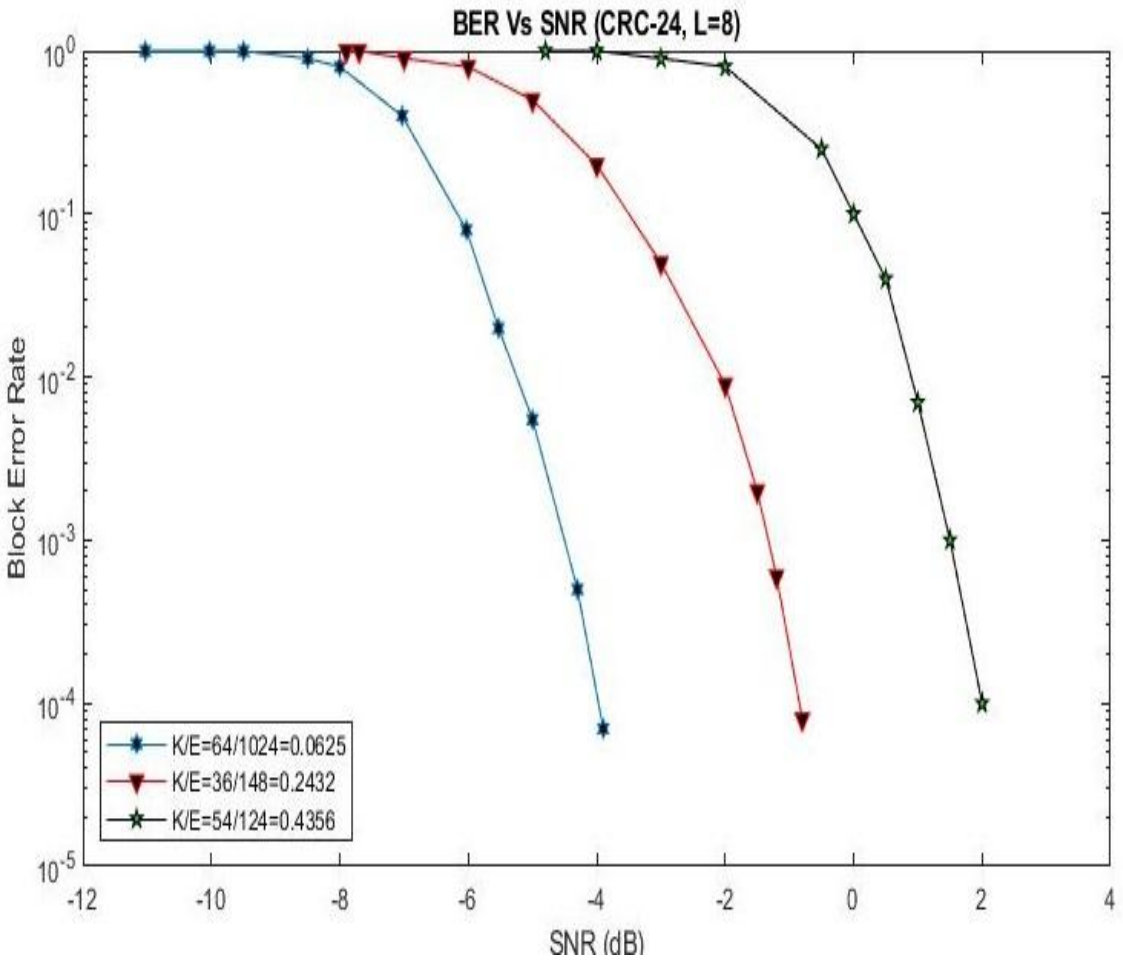


Figure 7.13: Block error rate vs. SNR (dB) for different code rate (CRC-24, L=8)

Table 7.9: Block error rate vs. SNR (dB) for different code Rate (CRC-24, L=8)

R= 0.0625			R= 0.2432			R= 0.4355	
SNR (dB)	BER		SNR (dB)	BER		SNR (dB)	BER
-11	1		-7.9	1		-4.8	1
-10	1		-7.7	1		-4	1
-9.5	1		-7	0.963321		-3	0.914952
-8.5	0.932745		-6	0.812483		-2	0.811465
-8	0.885412		-5	0.554176		-1	0.618851
-7	0.612454		-4	0.245484		0	0.178515
-6	0.081967		-3	0.052648		0.5	0.045512
-5.5	0.029631		-2	0.009364		1	0.007362
-4.8	0.005714		-1.5	0.002785		1.5	0.001476
-4.2	0.000725		-1.2	0.000637		2	0.000178
-3.8	0.000026		-0.8	0.000081		2.5	0.000013

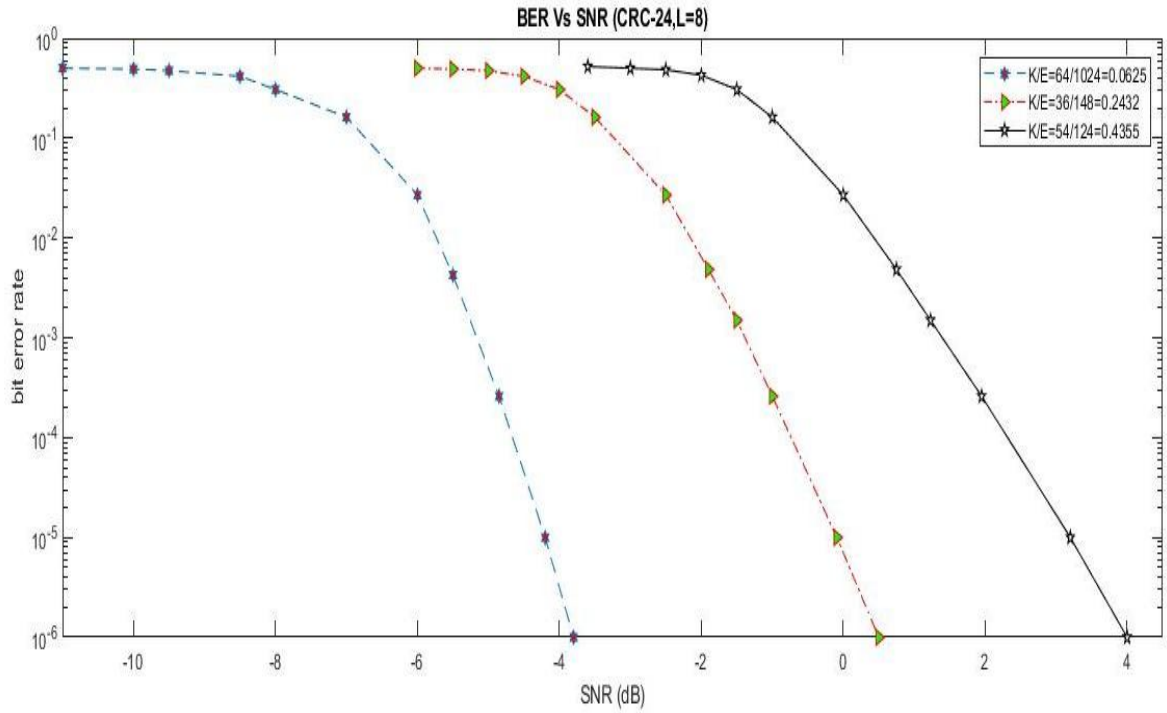


Figure 7.14: Bit error rate vs. SNR (dB) for different code rate (CRC-24, L=8)

Table 7.10: Bit error rate vs. SNR (dB) for different code Rate (CRC-24, L=8)

R= 0.0625			R= 0.2432			R= 0.4355	
SNR (dB)	BER		SNR (dB)	BER		SNR (dB)	BER
-11	0.5025134		-6	0.5025195		-3.6	0.5225012
-10	0.4925621		-5.5	0.4925714		-3	0.5025134
-9.5	0.4756879		-4.5	0.4175512		-2	0.4275914
-8.5	0.4175328		-4	0.3075198		-1.5	0.3075811
-7.85	0.3075913		-3.5	0.1625011		-1	0.1618252
-7	0.1625936		-2.5	0.0275205		0	0.0278515
-6	0.0274154		-1.8	0.0048573		0.75	0.0048512
-5.5	0.0042539		-1.5	0.0015375		1.2	0.0015662
-4.8	0.0002619		-1	0.0002856		2	0.0002476
-4.2	0.0000137		-0.09	0.0000197		3.2	0.0000178
-3.8	0.0000011		0.5	0.0000018		4	0.0000132

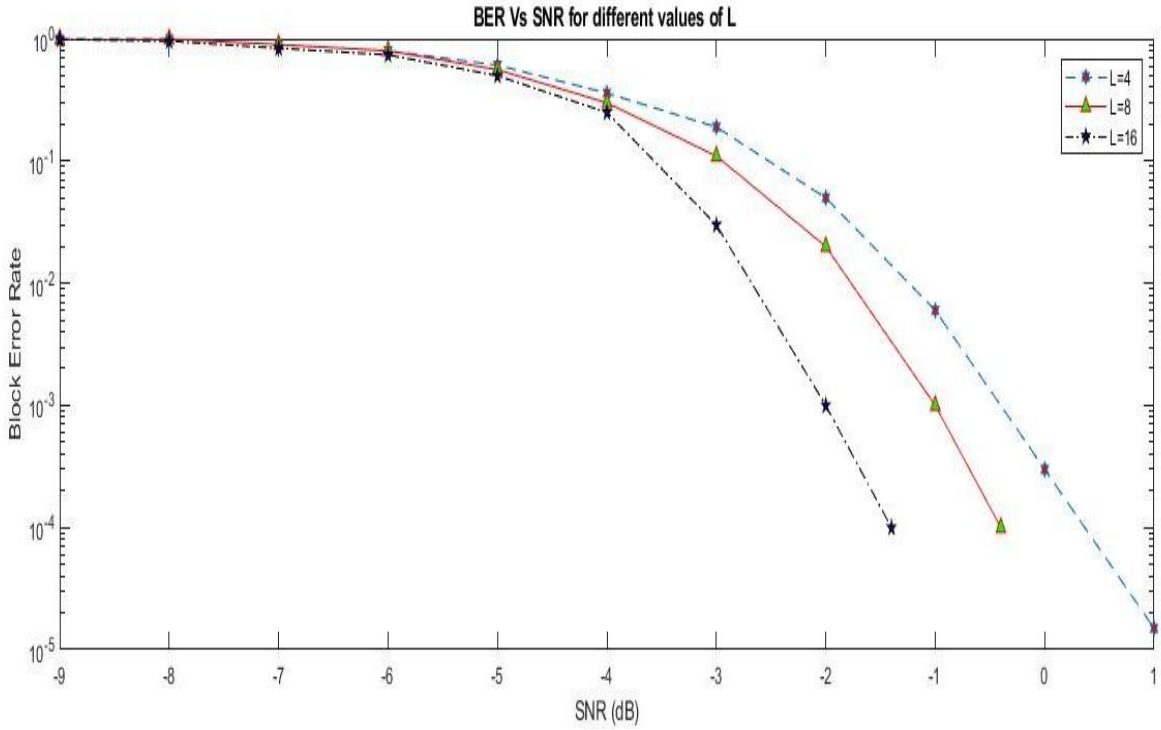


Figure 7.15: Block error rate vs. SNR (dB) for different List size (CRC-24, R=0.0625)

Table 7.11: Block error rate vs. SNR (dB) for different List size (CRC-24, R=0.0625)

L=4			L=8			L=16	
SNR (dB)	BER		SNR (dB)	BER		SNR (dB)	BER
-9	1		-9	1		-9	1
-8	1		-8	1		-8	0.994851
-7	0.907411		-7	0.901321		-7	0.864925
-6	0.802754		-6	0.772481		-6	0.721442
-5	0.615419		-5	0.554170		-5	0.488519
-4	0.361245		-4	0.305484		-4	0.228718
-3	0.181969		-3	0.112645		-3	0.054511
-2	0.059637		-2	0.031369		-2	0.001367
-1	0.006071		-1.3	0.000138		-1.6	0.000113
0	0.000625		-0.7	0.000237		-1	0.000015
1.2	0.000010		0	0.000010		-1.1	0.000000

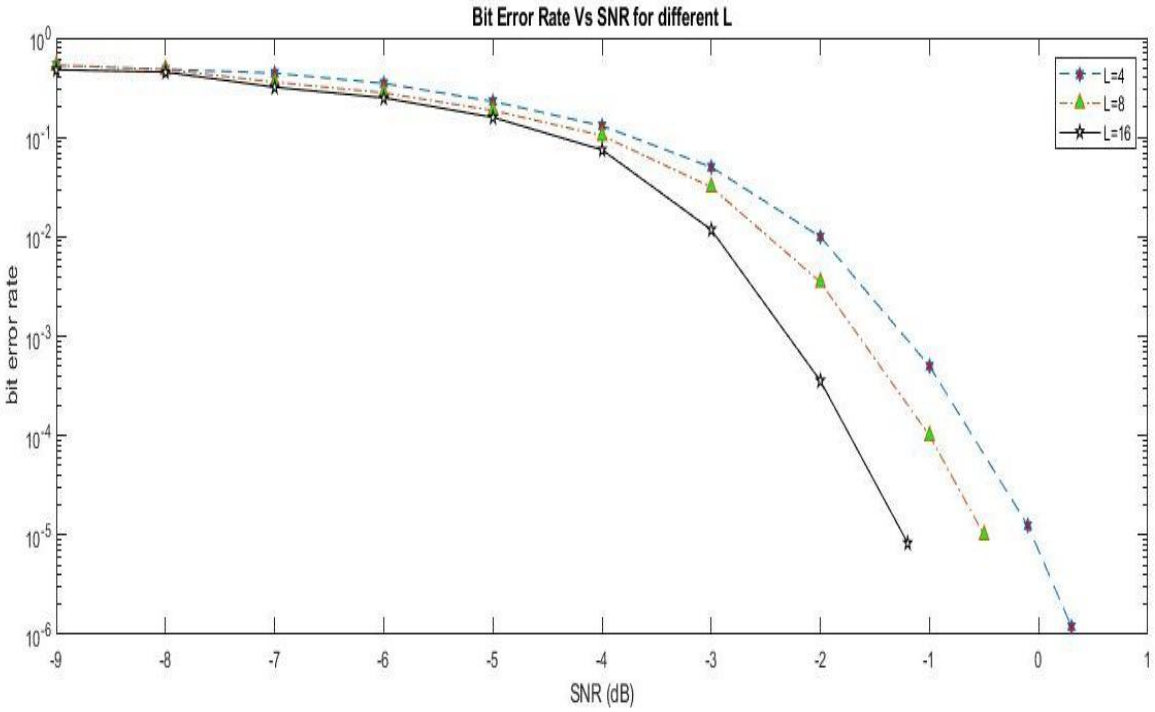


Figure 7.16: Bit error rate vs. SNR (dB) for different List size (CRC-24, R=0.0625)

Table 7.12: Bit error rate vs. SNR (dB) for different List size (CRC-24, R=0.0625)

L=4			L=8			L=16	
SNR (dB)	BER		SNR (dB)	BER		SNR (dB)	BER
-9	0.5533107		-9	0.5153061		-9	0.5016515
-8	0.4851329		-8	0.4492318		-8	0.4099481
-7	0.4411230		-7	0.3603441		-7	0.3086492
-6	0.3502754		-6	0.2745326		-6	0.2572144
-5	0.2361541		-5	0.1455417		-5	0.1088519
-4	0.1608612		-4	0.1030548		-4	0.0928712
-3	0.0903183		-3	0.0811264		-3	0.0145115
-2	0.0116372		-2	0.0071369		-2	0.0003671
-1	0.0050716		-1.3	0.0001381		-1.3	0.0000113
0	0.0000125		-0.7	0.0000137		-1	0.0000010
1.2	0.0000012		0	0.0000010		-1.1	0.0000000

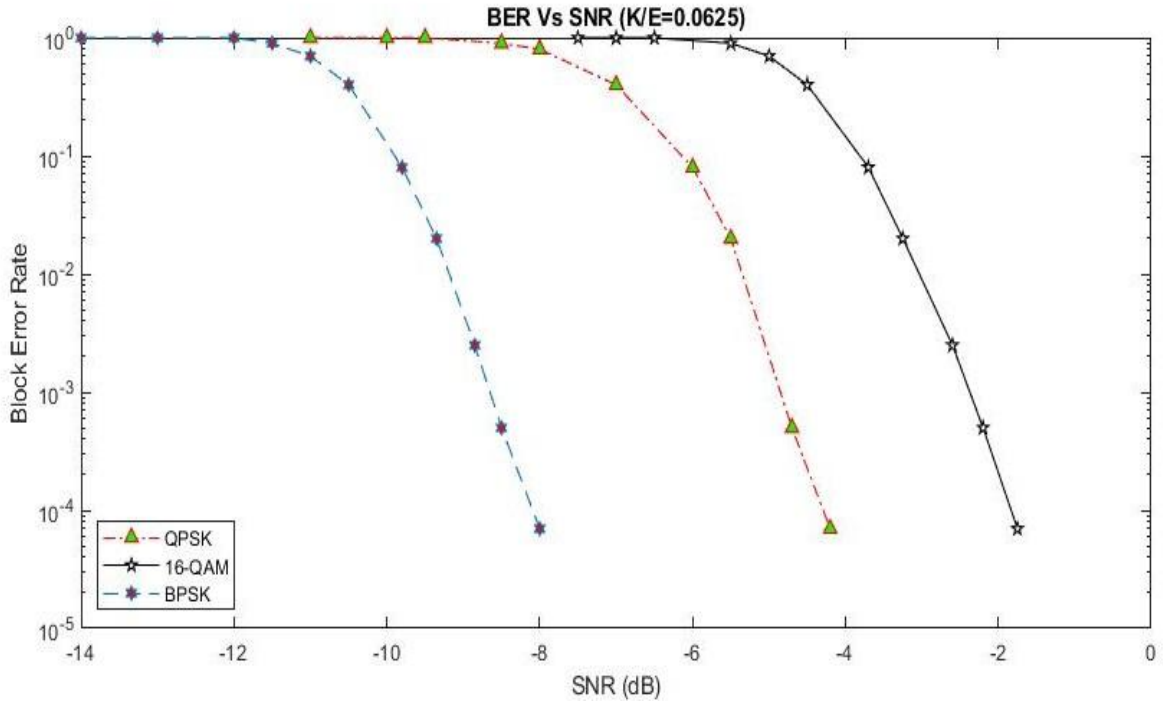


Figure 7.17: Block error rate vs. SNR (dB) for different modulation scheme (CRC-24, L=8, R=0.0625)

Table 7.13: Block error rate vs. SNR (dB) for different modulation scheme (CRC-24, L=8)

QPSK			BPSK			16-QAM	
SNR (dB)	BER		SNR (dB)	BER		SNR (dB)	BER
-11	1		-14	1		-7.5	1
-10	1		-13	1		-7	1
-9.5	1		-12	1		-6.5	1
-8.5	0.932745		-11.5	0.927153		-5.5	0.908114
-8	0.885412		-11	0.755417		-5	0.706188
-7	0.612454		-10.5	0.425484		-4.5	0.401785
-6	0.081967		-9.8	0.080648		-3.7	0.079551
-5.7	0.029631		-9.35	0.0266487		-3.25	0.024362
-4.8	0.000714		-8.85	0.0025785		-2.6	0.004761
-4.1	0.000072		-8.5	0.000537		-2.2	0.000507
-3.8	0.000011		-8	0.000083		-1.8	0.000076

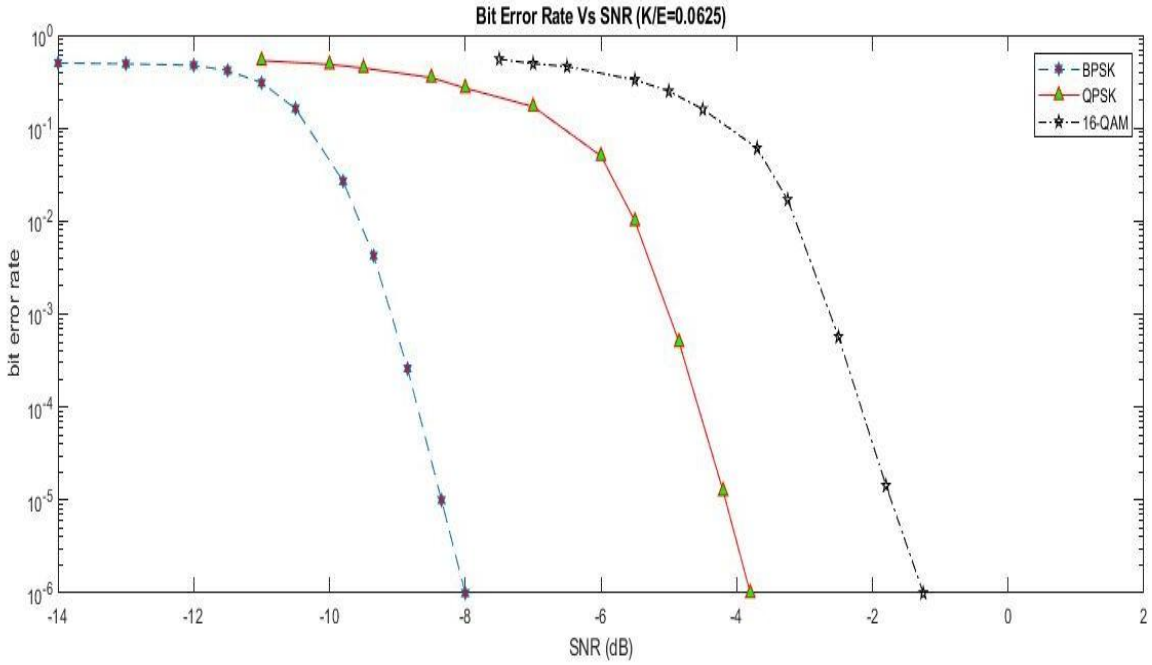


Figure 7.18: Bit error rate vs. SNR (dB) for different modulation scheme (CRC-24, L=8, R=0.0625)

Table 7.14: Bit error rate vs. SNR (dB) for different modulation scheme (CRC-24, L=8)

QPSK			BPSK			16-QAM	
SNR (dB)	BER		SNR (dB)	BER		SNR (dB)	BER
-11	0.5205124		-14	0.5025321		-7.5	0.5505179
-10	0.4925681		-13	0.4978648		-7	0.4535794
-9.5	0.4756879		-12	0.4868741		-6.5	0.3571254
-8.5	0.4175932		-11.5	0.4092715		-5.5	0.2790811
-8	0.3075885		-11	0.3275541		-5	0.1706188
-7	0.1624546		-10.5	0.2454894		-4.5	0.1040178
-6	0.0819675		-9.8	0.0806483		-3.7	0.0795519
-5.7	0.0296317		-9.35	0.0066487		-3.25	0.0243625
-4.8	0.0007142		-8.85	0.0003578		-2.6	0.0047616
-4.1	0.0000102		-8.5	0.0000053		-2.2	0.0001250
-3.8	0.0000010		-8	0.0000010		-1.8	0.0000010

7.3 Summary

In the first section of this chapter we have shown performance analysis [33] of polar codes for different conditions. First we have shown bit error rate and frame error rate for different code rate for a fixed code length. We have seen as the code rate R increase probability of bit error rate and frame error rate increase. Chances for errors in terms of bit error and frame error is higher in non-systematic code than systematic code. With the increasing code length, bit error rate and frame error rate decrease.

In the second section, performance analysis of CRC aided SCD is done. Here we have focused on block error rate and bit error rate for different conditions. We have seen that SNR (dB) increases with increasing code rate. Similarly chances of block error and bit error increase as code rate increase. List size plays an important role in CRC aided successive cancellation decoding. As list size increase probability of block error and bit error decrease. And finally we have shown performance of CRC aided polar with different modulation scheme. We have used BPSK, QPSK and 16-QAM modulations to observe the performance.

Chapter 8

Conclusion and Future work

8.1 Conclusion

Error-correcting codes play a crucial role in reliable and robust communication and storage system. The dream of researchers would be to achieve the channel capacity at low implementation complexity without compromising the latency and throughput requirements of communication systems. Polar codes are the latest class of modern error-correcting codes and they show high potential.

In the first part of our work, FPGA-based successive cancellation decoder has been implemented. By using common sub expressions, number of XOR operations are reduced which leads to compact decoder architecture. As code length increases latency and memory occupancy of decoder increase.

Second part is all about performance analysis of polar codes. With the increase of code length performance in terms of bit error and frame error rates improves with respect to Eb/No (dB). Similarly when code rate is increased then BER and FER increase. It is also found that BER and FER performance is better for systematic code than non-systematic code.

In case of CRC aided SCD, performance with respect to block and bit error rates vary with different code lengths, list size and also on modulation scheme. As list size increases the performance in terms of bit error rate and block error rate increase with SNR, means with increasing list size probability of error reduces. Three different modulation schemes namely BPSK, QPSK and 16-QAM have been used during simulation and it is found that for coded modulation, higher order modulation needs higher SNR for a fixed BER.

8.2 Future Work

Lots of works are on progress in the field of Channel Coding to flourish the data rates in reliable communication including upcoming 5^{th} generation technology. Performance of original SCD is not satisfactory for short code length, and it is a field where research can be done to improve the performance of the decoder even for short code length. Improving complexity and latency of successive cancellation and successive cancellation list decoder is an important area to work on. Concept of adaptive decoders is almost near where the same decoder will be able to perform well for both short and large code length. Lots of work is still left for CRC-aided successive cancellation decoder, search for best polynomial is still going on.

Some other fields which can be investigated are the effect of Bhattacharya parameter for the selection of frozen bits, the symmetric channel capacity, reliability of the channel and the effect of Polarization on the reliability, etc. Improving bit error rates and frame error rates is a matter of discussion where there is a trade-off between decoder complexity and latency.

References

[1] E. Arikan, "Channel polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memory less Channels", *IEEE Trans. Inform. Theory*, vol. 55, no. 7, July 2009.

[2] E. Arikan, "A Survey of Reed-Muller Codes from Polar Coding Perspective", *IEEE Inform. Theory Workshop on Information Theory*, Jan 2010.

[3] E. Arikan, "A Performance Comparison of Polar Codes and Reed-Muller Codes", *IEEE Communications Letters*, vol. 12, no. 6, June 2008.

[4] E. Arikan, H. Kim, G. Markarian, Ü. Özgür, E. Poyraz, "Performance of short polar codes under ML decoding, Proceedings of the ICT-Mobile", *Summit Conference*, June 2009.

[5] I. Tal, A. Vardy, "List Decoding of polar Codes", *IEEE International Symposium on Information Theory Proceedings*, 2011.

[6] I. Tal, A. Vardy, "List Decoding of polar Codes", *IEEE Transactions on Information Theory*, 61 (5), 2213 - 2226, 2015.

[7] G. Liva, L. Gaudio, T. Ninacs, T. Jerkovits, "Code Design for Short Blocks: A Survey", *arXiv preprint arXiv: 1610.00873*, Oct. 2016.

[8] H. Vangala, E. Viterbo, "A Comparative Study of Polar Code Constructions for the AWGN Channel", *arXiv*, Jan. 2015.

[9] M. Qin, J. Guo, A. Bhatia, A. Fàbregas, "Polar Code Constructions Based on LLR Evolution", *IEEE Communications Letters*, vol. 21, no. 6, June 2017

[10] B. Li, H. Shen, D. Tse, "A RM-Polar codes", *arXiv*, July 2014.

[11] M. Mondelli, S. H. Hassani, R. L. Urbanke, "From Polar to Reed-Muller Codes: A Technique to Improve the Finite-Length Performance", *IEEE Transactions on Communications*, vol. 62, no. 9, Sept. 2014.

[12] I. Dumer, K. Shabunov, "Soft Decision decoding of Reed-Muller codes: recursive lists", *IEEE Trans. Inform. Theory*, vol. 52, pp. 1260 to 1266, 2006.

[13] P. Koopman, T. Chakravarty," Cyclic Redundancy Code (CRC) Polynomial Selection for Embedded Networks", *Preprint: The International Conference on Dependable Systems and Networks*, DSN-2004.

[14] X. liang, Y. She, H. Vangala, X. You, C. Zhang, "An efficient successive cancellation polar decoder based on new folding approaches", *IEEE Trans*.

[15] O. Dizdar, E. Arikan, "A High Throughput Energy-efficient Implementation of Successive Cancellation Decoder for Polar Codes using Combinational Logic", *IEEE Trans*, vol-63,No-3, march 2016.

[16] "Early-trends-in-5g-technology-leadershiphttps://knect365.com/5g virtualisation/article/1a7b1504-12db-47ba-9ecf-aacad96c3582/early-trends-in-5g-technology-leadership"

[17] S. Mude, R. N. Bhukya, "Polar code and Polarization using Bhattacharya Parameter", *IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON*), pp-151-154, 2018.

[18] I. Tal and A. Vardy, "How to construct polar codes", *IEEE Trans. Inf. Theory*, vol. 59, no.10, pp. 6562–6582, Oct. 2013.

[19] E. Arıkan, "Channel polarization: a method for constructing capacity-achieving codes", *IEEE Int. Symp. On Inf. Theory (ISIT)*, pp. 1173–1177, Jul. 2008.

[20]H. Vangala, Y. Hong, E. Viterbo, "Efficient algorithms for systematic polar encoding."*IEEE communications letters*20, no. 1 (2016): 17-20

[21] E. Arıkan, "Systematic polar coding", *IEEE Communication Letter*, vol. 15, no. 8, pp. 860–862, 2011

[22] L. Huijum, "Polar codes: construction and performance improvement", *Electrical Engineering & Telecommunications, Faculty of Engineering,* UNSW.

[23]G. Sarkis, I. Tal, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, "Flexible and low-complexity encoding and decoding of systematic polar codes", *IEEE Trans. Commun., vol*. 64, no. 7, pp. 2732–2745, Jul. 2016

[24] Trifonov, Peter, "Efficient design and decoding of polar codes", *IEEE Transactions on Communications* 60, no. 11 (2012): 3221-3227.

[25] A. Eslami and H. Pishro-Nik, "A Practical Approach to Polar Codes",

[26] Kai Niu, Kai Chen, Jiaru Lin, and Q. T. Zhang, "Polar Codes: Primary Concepts and Practical Decoding Algorithms*", IEEE Communications Magazine*, July 2014.

[27] Stimming, A. B., Parizi, M. B., and Burg, A., "LLR-Based Successive Cancellation List Decoding of Polar Codes", *IEEE Transaction on Signal Processing*, vol. 63, No. 19, pp.5165-5179, 2015

[28] *https://in.mathworks.com/help/5g/gs/polar-coding*

[29] Niu, K., and Chen, K., "CRC-Aided Decoding of Polar Codes," *IEEE Communications Letters*, vol. 16, No. 10, pp. 1668-1671, Oct. 2012

[30] Qingshuang Zhang ; Aijun Liu ; Xiaofei Pan ; Kegang Pan, ”CRC Code Design for List Decoding of Polar Codes”, *IEEE Communications Letters*, vol-21, pp-1229-1232, 2017

[31] WenqingSong ;Chuan Zhang ; Shunqing Zhang ; Xiaohu You, ”Efficient adaptive successive cancellation list decoders for polar codes”, *IEEE International Conference on Digital Signal Processing (DSP)*, pp-218-222, 2016

[32] Wolf, W, “FPGA-Based System Design, Prentice-Hall Modern Semiconductor Design Series” *Prentice-Hall*, NJ, 2004

[33] N. Hussami, S. B. Korada, and R. Urbanke, “Performance of Polar Codes for Channel and Source Coding*,” Proceedings of IEEE International Symposium on Information Theory*, Seoul, Jul. 2009