Technische Universität München
Lehrstuhl für Nachrichtentechnik
Prof. Dr. sc. techn. Gerhard Kramer

Master's Thesis

Polar FEC chain development in Software for 5G

Vorgelegt von:
Yadhunandana Rajathadripura Kumaraiah

München, November 2018

Betreut von:
Dipl.-Ing. Advisor

Master's Thesis am
Lehrstuhl für Nachrichtentechnik (LNT)
der Technischen Universität München (TUM)
Titel : Polar FEC chain development in Software for 5G
Autor : Yadhunandana Rajathadripura Kumaraiah

Yadhunandana Rajathadripura Kumaraiah
Schröfelhofstraße 14 WG 02/02
81375 München
yadhu.kumaraiah@tum.de

Ich versichere hiermit wahrheitsgemäß, die Arbeit bis auf die dem Aufgabensteller bereits bekannte Hilfe selbständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderung entnommen wurde.

München, 12.09.2018
. . . .
Ort, Datum (Yadhunandana Rajathadripura Kumaraiah)

Contents

List of Figures

List of Tables

Abstract

The abstract comes here.

1 Introduction and Motivation

In 1948, scientists at the Bell Laboratories achieved two landmark research results: Claude E. Shannon published his paper *A mathematical theory of communication* and John Bardeen, Walter Brattain and William Shockley announced the invention of the *transistor effect*.

[1]

A binomial distribution is shown in Figure 1.1.

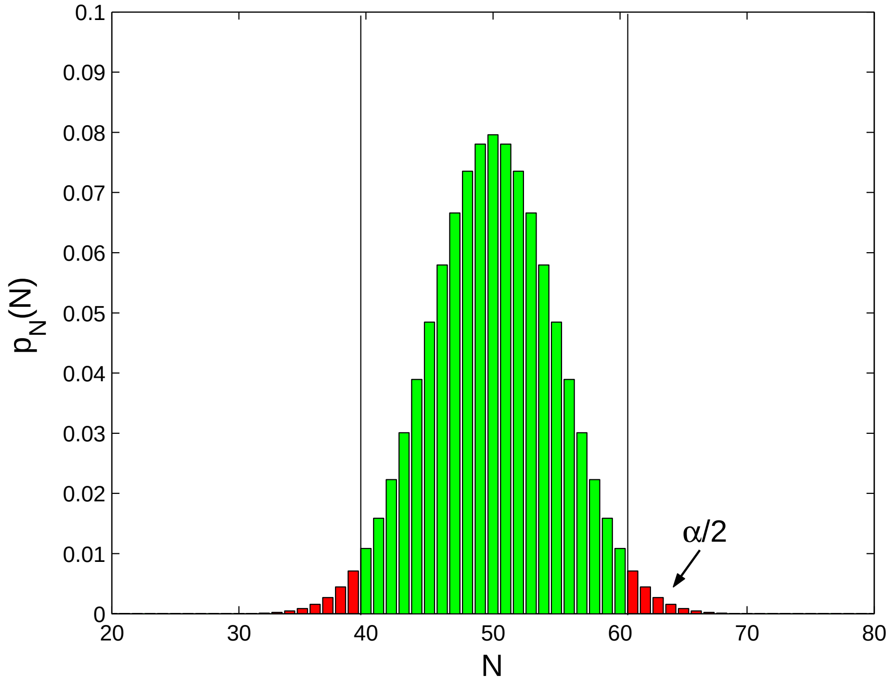

Figure 1.1: PDF $p_N(N)$ of the number N of times that the head side is up.

For further information, the reader is referred to [2].

Traditionally FEC chains are developed in hardware i.e FPGA's or ASIC's to achieve low latency and high throughput. Development in FPGA/hardware requires more time and costly. With recent advances in General Purpose Processors it is possible to achieve

required latency and throughput with software implementations without custom hardware. Software implementations are flexible and easy to maintain compared hardware implementations.
However algorithms need to be adopted/optimized to efficiently implement in software.
Recent advances in the modern processors such as SIMD units can be utilized to achieve low latency and high throughput.

2 Polar Codes Background

Polar codes were introduced by Arikan in his seminal work [1]. Polar codes are the class of capacity achieving codes. In the past decade, polar codes have sparked a interest from both academia and industrial resulting in significant research work in improving performance. The 5th generation wireless systems (5G) standardization has adopted polar codes for uplink and downlink control information for the enhanced mobile broadband (eMBB). Polar codes are also considered as the potential coding schemes for two other frameworks of 5G, namely ultra-reliable-low-latency (URLLC) and massive machine-type communications (mMTC).

Polar codes achieve capacity asymptotically for of memoryless channel. Although polar codes are the first theoretically capacity achieving codes with an explicit construction, capacity is approached only asymptotically their performance is suboptimal compared to LDPC or Turbo codes at short block lengths with successive cancellation decoding (SCD). [3] Presents the improved version of SCD called *successive cancellation list decoder(SCLD).*

The construction of polar codes involves the identification of channel reliability values, information bits are placed in the K high reliable bit indices out of N positions and remaining bits are set to zero then these N bits are passed through a polar encoding circuit to get the encoded bits. Selection is of reliability indices is done based on the code length and channel signal-to-noise ratio. Due to varying code length and channel conditions in 5G systems significant effort has been put to identify reliability indices which have good error correction performance over multiple code length and channel parameters.

2.1 Preliminaries

This section introduces the basic mathematical foundations of the polar codes. In particular, about the frozen set design, encoding and decoding. Different decoding algorithms are introduced. Mainly Successive Cancellation Decoding(SCD), Successive Cancellation

List Decoding(SCLD) and CRC aided Successive Cancellation List Decoding(CA-SCLD). Examples of encoding and decoding with different algorithms are presented for better understanding.

2.1.1 Polar codes definition

Mathematical foundations of polar codes lay on the polarization effect of the matrix [1]. $k = \begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}$ also called Arikan matrix. Polar codes are (N, K) linear block codes of size $N = 2^n$ where n being a natural number. N is the block length of the code and K is the number of information bits. N-bit vector U contains K information and $N - K$ frozen bits which are set to known value mostly zeros. These bits are then multiplied with the generator matrix constructed from Kronecker power of Arikan kernel matrix.

For example $n = 3$, block length N becomes 8 hence the generator matrix is

$$k^{\otimes 3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

where $k^{\otimes n}$ denotes the n^{th} Kronecker power of k. The encoding process involves the multiplication of N-bit vector U consisting of K information bits and $N - K$ frozen bits with $k^{\otimes n}$.

A Polar code construction

In polar coding, first step is to identify the channel reliability values for a particular block length, this step is also called as polar code construction. Basic idea of polar coding is to manufacture out of N independent copies of given binary discrete memoryless channel create a fraction of channels which are either completely noiseless or noisy.This process of creating extremal channels is called channel polarization. As $N \to \infty$ fraction of noiseless channels approaches capacity of channel. Estimating reliability indices channels is carried by considering the Bhattacharyya parameter which indicates the reliability of individual channel.

For a generic binary-input discrete memoryless channel (B-DMC) which is represented as $W\colon \mathcal{X} \to \mathcal{Y}$ with input alphabet $\mathcal{X}$, output alphabet $\mathcal{Y}$ and transition probabilities given by $W(y|x), x \in \mathcal{X}, y \in \mathcal{Y}$.
Bhattacharyya parameter is given by

$$Z(W) \triangleq \sum_{y \in \mathcal{Y}} \sqrt{W(y|0)W(y|1)} \tag{2.1}$$

Bhattacharyya parameter indicates how unreliable the channel is, It is easy see that $Z(W)$ takes values between $[0,1]$ better the channel smaller is $Z(W)$. Polarization creates channels with $Z(W)$ 0 or 1.
Here Give one example channel polarization for BEC channel and show the evolution of channel reliabilities. May include a insert a picture with evolution of channel reliabilities plot.

B Encoding

As explained in the section A code construction is done. Information bits are placed in the most reliable bit indices position non reliable bit positions are called frozen bits whose values are set to zeros. This N-bit vector U is multiplied with generator matrix obtained by the Kronecker power of Arikan kernel matrix. Multiplying with generator matrix can also be represented as circuit form. for $n = 3$ block length N becomes 8 for such a case encoding circuit looks as shown in the following figure.

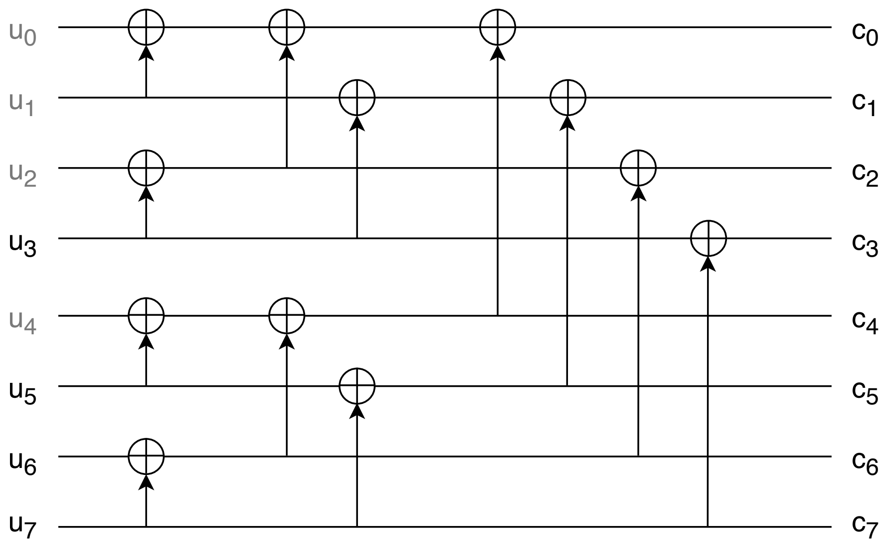

Figure 2.1: Polar encoder in circuit form for $N = 8$

the grayed locations are the frozen bit indices which are set to zero, in remaining positions information bits are inserted. Output of the circuit is a code word which is transmitted over the channel.

3 Encoding FEC Chain

Work I have done Until now.
Optimizations to the original implementations until now. Generic optimizations - Using optimization primitives such as likely and unlikely. - Aligning memory to 32 bytes so copying of data can be vectorized. Polar transform optimization - Replace binary additions with xor. instead of addition and then modulus two. - division and multiplications by left and right shift operations. - Avoided copy operations in polarTransform operations. Optimization in getting reliability indices. - Avoided remove and erase operations which have huge overhead. Wrote a efficient mechanism(reduced the latency by 176 us). - Instead removing and erasing I mark the element as removed. - Since the reliability indexes won't change. I built a look up table in place of searching all (1024)indices reduced the latency by 40us - Avoided copying operations of interleaved indexes. - Unrolled the loop to reduce the jumps. Rate matching optimizations. - optimization in subblock interleaving, Rewrote the logic to avoid E number of division and modulus operations. - Unrolled the for loops in subblock inteleaving method. - Implemented optimal version of bit selection, Avoided E number of modulus operations which are very costly. - Again optimization primitives for helping the branch predictor.
Fast version of Encoding API's. In the original implementation of the polar encoding each of the bit is treated as 32 bit integer. This is highly inefficient when the goal is to process multiple bits at time. With each bit considered as 32 bit integer SIMD instructions won't provide any performance improvement. Reason is SIMD instruction can process multiple bits at time. avx2 instructions 256bits at a time. if we have 32 bits to represent a single bit. we can process only 8 bits at time. Which doesn't significantly improve the performance. To avoid this disadvantage and make use of SIMD capability. each 64 bit integer is considered as 64 bits of data. so one avx2 instruction can process 256 data bits in a single instruction. - Built a look up table to avoid last eight stages of polar encoding instead of traversing till end of tree. - Implemented SIMD instruction based encoding. Encoding happens within 0.6 us for N = 512. - Implemented optimal version of CRC calculation which can calculate CRC for PDCCH chain within 0.8 us. Original implementation was taking 7 us. - Implemented a bit interleaver which can deal with this format of data.

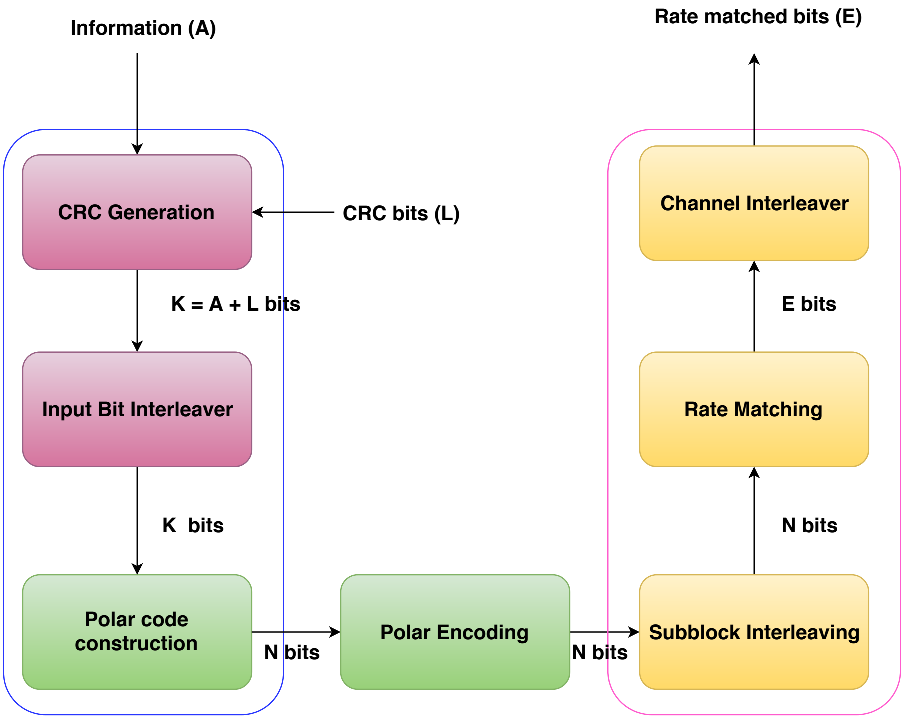

Figure 3.1: Polar Encoding FEC chain for PDCCH/PBCH

4 Decoding FEC Chain

Decoding is of serial nature, has lot of latency.

4.1 Decoding algorithms

5 Conclusion

Bibliography

[1] E. Arikan, “Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels,” in *IEEE Transactions on Information Theory*, vol. 55, pp. 3051–3073, 2009.

[2] T. Cover and J. Thomas, *Elements of Information Theory.* Wiley series in telecommunications, New York: John Wiley & Sons, 1991.

[3] I. Tal and A. Vardy, “List decoding of polar codes,” in *IEEE Transactions on Information Theory*, vol. 61, pp. 2213–2216, 2015.

[4] V. Bioglio, C. Condo, and I. Land, “Design of Polar Codes in 5G New Radio,” *ArXiv e-prints*, Apr. 2018.