Examensarbete

Implementation Aspects of 3GPP TD-LTE

Master thesis performed in
Computer Engineering
by
Ningning Guo

Report number: LiTH-ISY-EX--07/4122—SE
Linköping Date August 1st, 2009
Implementation Aspects of 3GPP TD-LTE

Master thesis in

Computer Engineering
Department of Electrical Engineering
At Linköping Institute of Technology

by

Ningning Guo

............................................................
LiTH-ISY-EX--07/4122--SE

Supervisor: Di Wu
ISY/Datorteknik, Linköpings universitet

Examiner: Dake Liu
ISY/Datorteknik, Linköpings universitet

Linköping 2009
Abstract

3GPP LTE (Long Term Evolution) is a project of the Third Generation Partnership Project to improve the UMTS (Universal Mobile Telecommunications System) mobile phone standard to cope with future technology evolutions. Two duplex schemes FDD and TDD are investigated in this thesis. Several computational intensive components of the baseband processing for LTE uplink such as synchronization, channel estimation, equalization, soft demapping, turbo decoding is analyzed. Cost analysis is hardware independent so that only computational complexity is considered in this thesis. Hardware dependent discussion for LTE baseband SDR platform is given according the analysis results.

Keywords:
3GPP LTE, FDD, TDD, OFDM, SC-FDMA, MIMO, FFT, IFFT, SDR
Abstract

3GPP LTE (Long Term Evolution) is a project of the Third Generation Partnership Project to improve the UMTS (Universal Mobile Telecommunications System) mobile phone standard to cope with future technology evolutions. Two duplex schemes FDD and TDD are investigated in this thesis. Several computational intensive components of the baseband processing for LTE uplink such as synchronization, channel estimation, equalization, soft demapping, turbo decoding is analyzed. Cost analysis is hardware independent so that only computational complexity is considered in this thesis. Hardware dependent discussion for LTE baseband SDR platform is given according the analysis results.

Keywords: 3GPP LTE, FDD, TDD, OFDM, SC-FDMA, MIMO, FFT, IFFT, SDR
Acknowledgement

First and foremost I wish, in these lines to thank for all the people and mainly, Di Wu for his stimulating suggestions and encouragement in all the time of the thesis work as my supervisor. In addition, grateful thanks are due to Professor Dake Liu for constructive comments and valuable suggestions during my study in Linköpings University.

I also wish to thank all the friends in Linköping for always making me feel welcome and feel less far from home.

Last but not least, I would like to thank my wonderful family, including my parents, my wife Yi Shi for all her love and support, my brothers and sister, for all their enduring support and always believing on me.
# Contents

## CHAPTER 1 INTRODUCTION ........................................................................................................ 1

1.1 BACKGROUND .......................................................................................................................... 1  
1.1.1 Software Defined Radio ....................................................................................................... 2  
1.1.2 Components in Base stations ............................................................................................ 2  
1.2 PURPOSE OF THE THESIS ..................................................................................................... 3  
1.3 OUTLINE .................................................................................................................................. 3  

## CHAPTER 2 OVERVIEW OF 3GPP LTE ..................................................................................... 4

2.1 INTRODUCTION .......................................................................................................................... 4  
2.1.1 Design Goals & parameters ............................................................................................... 5  
2.2 LTE BASIC CONCEPTS .............................................................................................................. 5  
2.2.1 Sub-Carrier .......................................................................................................................... 5  
2.2.2 Orthogonal Frequency Division Multiplexing (OFDM) .......................................................... 5  
2.2.3 Single Carrier with Frequency Domain Equalization (SC/FDE) ......................................... 6  
2.2.4 Cyclic Prefix (CP) ............................................................................................................. 6  
2.2.5 SC-FDMA and OFDMA ....................................................................................................... 7  
2.2.6 Smart antenna techniques ................................................................................................. 8  
2.3 LTE PHYSICAL LAYER .............................................................................................................. 8  
2.3.1 Generic Frame Structure .................................................................................................... 9  
2.3.2 Uplink .................................................................................................................................. 9  
2.3.3 Multiplexing ....................................................................................................................... 11  
2.3.4 Physical Uplink Shared Channels ...................................................................................... 11  
2.3.5 Uplink Reference Signal ................................................................................................... 11  

## CHAPTER 3 TD-LTE AND FDD LTE ......................................................................................... 12

3.1 FRAME STRUCTURE ............................................................................................................... 12  
3.2 FEATURES ROOTED FROM FRAME STRUCTURES ................................................................... 12  
3.3 ADVANTAGES AND DRAWBACKS ........................................................................................ 14  

## CHAPTER 4 COMPUTATIONAL COMPLEXITY ANALYSIS ......................................................... 16

4.1 OVERALL SYSTEM FLOW ....................................................................................................... 16  
4.1.1 LTE Downlink .................................................................................................................... 17  
4.1.2 LTE Uplink ........................................................................................................................ 20  
4.2 COMPLEXITY ANALYSIS FOR LTE SUPPORTED FFT/IFFT .................................................. 21  
4.2.1 Fast Fourier transform and Inverse FFT ........................................................................... 22  
4.2.2 Radix-2 FFT ..................................................................................................................... 22  
4.2.3 Radix-4 FFT ..................................................................................................................... 24  
4.1.4 Split-Radix FFT ............................................................................................................... 28  
4.2.5 Radix-3, Radix-5 and Radix-r FFT ................................................................................. 28
List of Figures

Figure 1.1 Current-generation SDR architecture
Figure 2.1 3GPP Evolution Flow
Figure 2.2 Block diagram of SC/FDE and OFDM
Figure 2.3 Cyclic prefix attached to the front of two successive symbols
Figure 2.4 overview structures of SC-FDMA and OFDMA
Figure 2.5 Subcarrier mapping schemes of SC-FDMA
Figure 2.6 Differences between OFDMA and SC-FDMA
Figure 2.7 LTE generic frame structure shared by both UL and DL
Figure 2.8 LTE Physical Resource Blocks structure
Figure 2.9 Overview of uplink physical channel processing
Figure 3.1 (a) Frame structure type1 FDD (b) Frame structure type2 TDD
Figure 4.1 Downlink system model for LTE
Figure 4.2 LTE downlink pilot symbol structure
Figure 4.3 Uplink system model for LTE
Figure 4.4 Butterfly computation structure (a) DIT FFT (b) DIF FFT
Figure 4.5 8-point radix-2 DIT FFT algorithm
Figure 4.6 Radix-4 DIT FFT butterfly computation structure
Figure 4.7 A recursive decomposing method for DFT calculation
Figure 4.8 16-point Radix-4 DIT FFT algorithm
Figure 4.9 Split-radix FFT butterfly
Figure 4.10 12 point Mix-radix Divide & Conquer FFT algorithm
Figure 4.11 Hybrid frequency/time domain PRACH generation
Figure 4.12 PRACH receiver structure
Figure 4.13 Signature Detection based on Power Delay Profile computation
Figure 4.14 LTE uplink block-type pilot structure
Figure 4.15 Frequency domain linear interpolation
Figure 4.16 16-QAM Gray-labeled Constellation
Figure 4.17 Turbo encoder structure
Figure 4.18 Structure of rate 1/3 turbo encoder
Figure 4.19 An iterative Turbo decoder
Figure 5.1 Base Station Architecture
Figure 5.2 eNodeB Baseband High Level Architecture of ARICENT solutions
Figure 5.3 DSP/FPGA partitioning for LTE uplink SC-FDMA systems
List of Tables

Table 1.1 Specifications release from 3GPP
Table 2.1 Evolution of mobile telecommunication technology
Table 2.2 LTE design parameters
Table 2.3 Uplink SC-FDE Modulation Parameters
Table 3.1 Uplink-downlink configurations for TDD subframe
Table 3.2 Main features derived from TDD frame structure
Table 3.3 Advantages / disadvantages of LTE TDD and LTE FDD
Table 4.1 LTE downlink/uplink N-point FFT size
Table 4.2 LTE uplink Transform Precoding M-point DFT size
Table 4.3 Radix-r FFT complexity
Table 4.4 N-point Radix-r FFT complexity
Table 4.5 Mixed-radix Divided and Conquer DFT complexity in MACs and Flops
Table 4.6 Random access preamble format
Table 4.7 Conversion from complex to real operations
Table 4.8 Complexity Analysis result for LTE uplink FFT-based LS channel estimation
Table 4.9 LLR Approximation for 4-QAM Gray-Coded Constellations
Table 4.10 LLR Approximation for 16-QAM Gray-Coded Constellations
Table 4.11 LLR Approximation for 64-QAM Gray-Coded Constellations
Table 4.12 Number of equivalent additions per operation
Table 4.13 Complexity of M-QAM per symbol in equivalent additions
Table 4.14 Complexity of M-QAM per frame in equivalent additions
Table 4.15 Complexity for turbo decoding with rate 1/2
Table 5.1 Differences among ASIC, FPGA and DSP
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>3GPP</td>
<td>3rd Generation Partnership Project</td>
</tr>
<tr>
<td>ADC</td>
<td>Analog to Digital Converter</td>
</tr>
<tr>
<td>ASIC</td>
<td>Application Specific Integrated Circuits</td>
</tr>
<tr>
<td>ASIP</td>
<td>Application Specific Instruction-set Processor</td>
</tr>
<tr>
<td>CDMA</td>
<td>Code Division Multiple Access</td>
</tr>
<tr>
<td>CP</td>
<td>Cyclic Prefix</td>
</tr>
<tr>
<td>DAC</td>
<td>Digital to Analog Converter</td>
</tr>
<tr>
<td>DDC</td>
<td>Digital Down Converter</td>
</tr>
<tr>
<td>DFE</td>
<td>Digital Front End</td>
</tr>
<tr>
<td>DFT</td>
<td>Discrete Fourier Transform</td>
</tr>
<tr>
<td>DSP</td>
<td>Digital Signal Processor</td>
</tr>
<tr>
<td>DwPTS</td>
<td>Downlink Pilot Time Slot</td>
</tr>
<tr>
<td>DUC</td>
<td>Digital Up Converter</td>
</tr>
<tr>
<td>eHSPA</td>
<td>evolved High-Speed Packet Access</td>
</tr>
<tr>
<td>FDD</td>
<td>Frequency Division Duplex</td>
</tr>
<tr>
<td>FEC</td>
<td>Forward Error Correction</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier Transform</td>
</tr>
<tr>
<td>FPGA</td>
<td>Field Programmable Gate Array</td>
</tr>
<tr>
<td>GP</td>
<td>Guard Period</td>
</tr>
<tr>
<td>GSM</td>
<td>Global System for Mobile Communications</td>
</tr>
<tr>
<td>HARQ</td>
<td>Hybrid Automated Repeat Request</td>
</tr>
<tr>
<td>HR</td>
<td>Hardware Radio</td>
</tr>
<tr>
<td>HSDPA</td>
<td>High Speed Downlink Packages Access</td>
</tr>
<tr>
<td>ICI</td>
<td>Inter Carrier Interference</td>
</tr>
<tr>
<td>IFFT</td>
<td>Inverse Fast Fourier Transform</td>
</tr>
<tr>
<td>ISR</td>
<td>Ideal Software Radio</td>
</tr>
<tr>
<td>ISI</td>
<td>Inter Symbol Interference</td>
</tr>
<tr>
<td>JTRS</td>
<td>Joint Tactical Radio System</td>
</tr>
<tr>
<td>LS</td>
<td>Least Squares</td>
</tr>
<tr>
<td>LTE</td>
<td>Long Term Evolution</td>
</tr>
<tr>
<td>MIMO</td>
<td>Multiple Input Multiple Output</td>
</tr>
<tr>
<td>OFDM</td>
<td>Orthogonal Frequency Division Multiplexing</td>
</tr>
<tr>
<td>OFDMA</td>
<td>Orthogonal Frequency Division Multiple Access</td>
</tr>
<tr>
<td>PBCH</td>
<td>Physical Broadcast Channel</td>
</tr>
<tr>
<td>PDCCH</td>
<td>Physical Downlink Control Channel</td>
</tr>
<tr>
<td>PDSCH</td>
<td>Physical Downlink Shared Channel</td>
</tr>
<tr>
<td>PHICH</td>
<td>Physical Hybrid-ARQ Indicator Channel</td>
</tr>
<tr>
<td>PRACH</td>
<td>Physical Random Access Channel</td>
</tr>
<tr>
<td>PRBs</td>
<td>Physical Resource Blocks</td>
</tr>
<tr>
<td>PUCCH</td>
<td>Physical Uplink Control Channel</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>PUSCH</td>
<td>Physical Uplink Shared Channel</td>
</tr>
<tr>
<td>RTT</td>
<td>Round-Trip Time</td>
</tr>
<tr>
<td>SC-FDMA</td>
<td>Single Carrier-Frequency Division Multiple Access</td>
</tr>
<tr>
<td>SCR</td>
<td>Software-Controlled Radios</td>
</tr>
<tr>
<td>SDR</td>
<td>Software Defined Radio</td>
</tr>
<tr>
<td>SINR</td>
<td>Signal to Interference and Noise Ratio</td>
</tr>
<tr>
<td>SISO</td>
<td>Single Input Single Output / Soft Input Soft Output</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal to Noise Ratio</td>
</tr>
<tr>
<td>TDD</td>
<td>Time Division Duplex</td>
</tr>
<tr>
<td>UE</td>
<td>User Equipment</td>
</tr>
<tr>
<td>UMTS</td>
<td>Universal Mobile Telecommunications System</td>
</tr>
<tr>
<td>UpPTS</td>
<td>Uplink Pilot Time Slot</td>
</tr>
<tr>
<td>USR</td>
<td>Ultimate Software Radios</td>
</tr>
<tr>
<td>WCDMA</td>
<td>Wideband Code Division Multiple Access</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Background

Currently, the worldwide UMTS networks are being upgraded to High Speed Downlink Packet Access (HSDPA) in order to increase the data rate and the capacity for downlink packet data. Meanwhile, concepts for UMTS Long Term Evolution (LTE) have been investigated to achieve more advanced goal. According to 3GPP LTE standard, OFDM (Orthogonal Frequency Division Multiplexing), SC-FDMA (single carrier – Frequency Division Multiple Access) and MIMO (Multiple Input Multiple Output) are the main technologies involved. The standardization for LTE is almost finalized by now, future changes in the specification are mainly bug fixes. Base on these standards listed in Table 1.1, the baseband design becomes more and more complex. Many techniques used for earlier GSM (FDMA/TDMA based), CDMA and HSPA (CDMA based) systems do not meet the performance and latency requirements of the LTE system any more. The problem is solved by building a multi-mode, multi-band, multi-functional mobile base station which is called Software Defined Radio (SDR).

<table>
<thead>
<tr>
<th>Version</th>
<th>Released Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 98</td>
<td>early 1999</td>
<td>Specify pre-3G GSM networks with previous releases (TDMA/FDMA based)</td>
</tr>
<tr>
<td>Release 99</td>
<td>early 2000</td>
<td>Specified the first UMTS 3G networks with a CDMA air interface (CDMA based)</td>
</tr>
<tr>
<td>Release 4</td>
<td>Mar. 2001</td>
<td>Added features including an all-IP Core Network</td>
</tr>
<tr>
<td>Release 7</td>
<td>Dec. 2007</td>
<td>Focus on decreasing latency, improvements to QoS and real-time applications such as VoIP. Also focus on OFDM techniques in downlink</td>
</tr>
<tr>
<td>Release 8</td>
<td>Dec. 2008</td>
<td>Specify LTE (OFDM-MIMO based)</td>
</tr>
<tr>
<td>Release 9</td>
<td>In progress, expected to be frozen in Dec. 2009</td>
<td>SAES Enhancements, WiMax and LTE/UMTS Interoperability</td>
</tr>
<tr>
<td>Release 10</td>
<td>In progress</td>
<td>Specify LTE Advanced</td>
</tr>
</tbody>
</table>

Table 1.1 Specifications release from 3GPP [1]
1.1.1 Software Defined Radio

The SDR technology is first promoted by U.S. military project named SpeakEasy to use programmable processing to emulate the existing military radios. In 1997, the Joint Tactical Radio System (JTRS) project has been created for US government and NATO to provide flexible and interoperable communication radios. JTRS project replaced approximately 750,000 military transceivers with 250,000 SDR radios [11].

According to SDR Forum [11] (International organization for promoting development and use of SDR technologies), there are five groups of software-radio categories: Tier 0 Hardware radio (HR), Tier 1 Software Controlled Radios (SCR), Tier 2 Reconfigurable SDRs, Tier 3 Ideal Software Radio (ISR) and Tier 4 - Ultimate Software Radios (USR). Among these groups, Tier 2 Reconfigurable SDRs are most commonly used technology nowadays. They provide software control of a variety of modulation schemes, wide or narrow band operation, communication security functions (such as hopping), and waveform requirements of current and future standards over a large frequency range. Figure 1.1 shows a current generation SDR system.

![Figure 1.1 Current-generation SDR architecture [12]](image)

1.1.2 Components in Base stations

Here lists the main components in Base stations:

- ADC: The single most demanding performance
- DAC: similar to ADC requirements
DDC / DUC: (Digital down/up converters) programmable embedded DSP functionalities with NCO for RSP/TSP with frequency hopping capability
FPGAs: Embedded DSP functionalities
DSP: meet computational requirements of base band processing
Host Processor: OE, Protocol stacks, MMI, system controls
Operating System S/W and F/W
Broadband RF Front End
Smart Antennas
Multi-Carrier Power Amplifier (MCPA)

This thesis will only focus on LTE baseband physical layer processing and the baseband architecture is briefly introduced in chapter 5.

1.2 Purpose of the Thesis

The main purpose of this thesis is to study the computational complexity of the 3GPP LTE uplink baseband processing at eNodeB (LTE base station) side, find out the difference between two duplex schemes (FDD and TDD), and give a brief discussion about hardware design targeting software-defined base station based on the complexity analysis results. For convenience, a simple scenario with 20MHz bandwidth and slow fading channel is considered for the analysis.

1.3 Outline

Chapter 2 first introduces the basic concepts of LTE including OFDM, OFDMA, SC-FDMA and MIMO. Then it comes to the brief descriptions of the Physical layer for Uplink and downlink.

Chapter 3 will briefly discuss the difference in baseband between TD-LTE and LTE FDD.

Chapter 4 studies several key algorithms used in the LTE system and give the computation complexity of the algorithms involved, such as FFT/IFFT, uplink synchronization, channel estimation, demodulating (including equalization and soft demapping) and turbo decoding.

Chapter 5 will give an brief introduction about nowadays radio base station and then discusses about the hardware requirements according to the complexity analysis result from chapter 4. The design consideration is brief discussed according to the 3GPP LTE standard

Chapter 6 gives the conclusion and future possible works are disclosed at the end.
Chapter 2

Overview of 3GPP LTE

2.1 Introduction

Long Term Evolution (LTE) is the next generation mobile telecommunication technology (Figure 2.1). According to the standard, LTE provides an uplink speed of up to 50 megabits per second (Mbps) and a downlink speed of up to 100 Mbps. No doubt, LTE will bring many benefits to cellular networks (Table 2.1). The bandwidth of LTE is from 1.4 MHz to 20 MHz [2]. The network operators may choose different bandwidth and provide different services based on the spectrum. It is also the design goal to improve spectral efficiency in 3G networks, allowing carriers to provide more data packets over a given bandwidth.

<table>
<thead>
<tr>
<th></th>
<th>WCDMA(UMTS)</th>
<th>HSPA (HSDPA/HSUPA)</th>
<th>HSPA+</th>
<th>LTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Downlink max speed (bps)</td>
<td>384k</td>
<td>14M</td>
<td>28M</td>
<td>100M</td>
</tr>
<tr>
<td>Uplink max speed (bps)</td>
<td>128k</td>
<td>5.7M</td>
<td>11M</td>
<td>50M</td>
</tr>
<tr>
<td>Latency - RTT</td>
<td>150ms</td>
<td>100ms</td>
<td>50ms(max)</td>
<td>~10ms</td>
</tr>
<tr>
<td>3GPP release</td>
<td>Rel 99/4</td>
<td>Rel 5/6</td>
<td>Rel 7</td>
<td>Rel 8</td>
</tr>
<tr>
<td>Access methodology</td>
<td>CDMA</td>
<td>CDMA</td>
<td>CDMA</td>
<td>OFDMA/SC-FDMA</td>
</tr>
</tbody>
</table>

Table 2.1 Evolution of mobile telecommunication technology [3]

- The Round Trip Time (RTT) is the latency from the UE throughput the channel to the BS and back

Figure 2.1 3GPP Evolution Flow [4]
Technical specifications for 3GPP LTE are not yet finalized, more details are emerging. This master thesis will only focus on physical layer (PHY).

### 2.1.1 Design Goals & parameters

The objective of LTE is to achieve high-data-rate, low-latency and packet-optimized radio-access. The LTE PHY is designed to support flexible transmission bandwidth up to 20MHz with the introduction of new transmission schemes and smart antenna technologies [5]. The design parameters are listed in Table 2.2.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel bandwidths (MHz)</td>
<td>1.4, 3, 5, 10, 15, 20</td>
</tr>
<tr>
<td>Modulation types supported</td>
<td>QPSK, 16QAM, 64QAM</td>
</tr>
<tr>
<td>Peak downlink speed 64QAM(Mbps)</td>
<td>100(SISO), 172(2x2 MIMO), 326(4x4 MIMO)</td>
</tr>
<tr>
<td>Peak uplink speed (Mbps)</td>
<td>50 (QPSK), 57 (16QAM), 86 (64QAM)</td>
</tr>
<tr>
<td>MIMO configurations</td>
<td>Downlink:4x2, 2x2, 1x2, 1x1</td>
</tr>
<tr>
<td></td>
<td>Uplink:1x2, 1x1</td>
</tr>
<tr>
<td>Spectrum efficiency</td>
<td>Downlink: 3 to 4 times HSDPA Rel.6</td>
</tr>
<tr>
<td></td>
<td>Uplink: 2 to 3 times HSUPA Rel.6</td>
</tr>
<tr>
<td>Latency</td>
<td>Idle to active less than 100ms</td>
</tr>
<tr>
<td></td>
<td>Small packets ~10ms</td>
</tr>
<tr>
<td>mobility</td>
<td>0-15km/h (optimized), 15-120km/h (high performance), 500/km/h (maximum)</td>
</tr>
<tr>
<td>coverage</td>
<td>Full performance up to 5km, Slight degradation 5km to 30km</td>
</tr>
<tr>
<td></td>
<td>Operation up to 100 km should not be precluded by standard</td>
</tr>
</tbody>
</table>

**Table 2.2 LTE design parameters [5], [10]**

### 2.2 LTE Basic Concepts

#### 2.2.1 Sub-Carrier

A sub-carrier is a narrow band carrier for use in OFDM based communications. Sub-carriers will be spread over the frequency baseband allocated to the user creating a spectrum of up to 1200 narrow band and orthogonal carriers.

#### 2.2.2 Orthogonal Frequency Division Multiplexing (OFDM)

Frequency-division multiplexing (FDM) is a form of signal multiplexing where multiple baseband signals are modulated on different frequency sub-carriers and composited into one signal. Orthogonal Frequency Division Multiplexing (OFDM) is based on FDM and utilizes orthogonal sub-carriers to transmit data. Compared to single carrier systems relying on increased symbol rates for higher data rates, OFDM systems divide the available bandwidth into many narrower sub-carriers and transmit data in parallel streams.
OFDM is the main technology for 3GPP LTE Downlink. The main advantages of OFDM are low complexity for implementation and high spectral efficiency, whereas high Peak-to-Average Power Ratio (PAPR) and high sensitivity to frequency offset are the main drawbacks.

### 2.2.3 Single Carrier with Frequency Domain Equalization (SC/FDE)

The single carrier modulated signals with frequency domain equalization has been known since the early 1970’s. Single carrier with frequency domain equalization (SC/FDE), combining Fast Fourier Transform (FFT) processing and cyclic prefix techniques, have the similar low complexity as OFDM systems.

From **figure 2.2** we can see the similar structure of OFDM and SC/FDE. The only difference is the position of IDFT. It is also called DFTS-OFDM. The main advantages of SC-FDE system are lower PAPR, lower sensitivity to carrier frequency offset and similar complexity in the receiver with lower complexity in the transmitter, which will benefit the UE, compared to OFDM system.

### 2.2.4 Cyclic Prefix (CP)

A cyclic prefix is a copy of the last part of a symbol attached to the beginning. CP provides a guard time between two successive symbols. If the length of a CP is longer than the maximum spread delay of the channel, there will be no ISI (Inter Symbol Interference) which means two successive symbols will not interfere with each other. It also avoids the ICI (Inter Carrier Interference) between sub-carriers because it uses a copy of the last part of the symbol.
Two types of CP, normal and extended CP are supported in LTE depending on the channel delay spread.

2.2.5 SC-FDMA and OFDMA

Making more efficient use of network resources, SC-FDMA (Single Carrier-Frequency Division Multiple Access) and OFDMA (Orthogonal Frequency Division Multiple Access) are used for multiplexing resources to multi-users in uplink and downlink respectively. Similar to OFDM and SC/FDE, OFDMA and SC-FDMA have similar structures. SC-FDMA can be seen as a DFT spread OFDMA system. Distributed and localized subcarrier mapping schemes can be used after IDFT process (Figure 2.5).

Figure 2.4 overview structure of SC-FDMA and OFDMA [6]

Figure 2.5 Subcarrier mapping schemes of SC-FDMA [6]
(Transmitted symbols are in the time domain for N=4 subcarriers per user, Q=3 users, and M=12 subcarriers in the system.)
The difference between OFDMA and SC-FDMA is that SC-FDMA has an IDFT processing before detection, which makes it less sensitive to a null in the channel spectrum. Furthermore, compared to OFDMA sending different symbols simultaneously, Figure 2.6 shows that SC-FDMA divides symbols into small blocks and transmit them in the order according to which subcarrier mapping scheme is implemented.

2.2.6 Smart antenna techniques

MIMO (Multiple Input Multiple Output) is one of several forms of smart antenna technology. It uses multiple antennas at both the transmitter and receiver side to improve the communication performance. MIMO technology brings significant improvement in data throughput and link range without additional bandwidth or transmit power. It achieves this by higher spectral efficiency (more bits per second per hertz of bandwidth) and link reliability or diversity (reduced fading) [7]. The high data throughput is achieved by using spatial multiplexing, while spatial diversity provides high link reliability. From encoding point of view, two types of encoding method can be used for MIMO system which are open-loop and closed-loop approach. The difference between open-loop and closed-loop is that closed-loop approach requires channel information and using weights computed from the channel estimation to perform precoding.

Closed-loop spatial multiplexing and open-loop with or without CCD for transmit diversity MIMO encoding schemes are adopted for LTE downlink. For LTE uplink, only one TX antenna is used during the transmission [8], so SIMO system is adopted for LTE uplink and only open-loop spatial multiplexing is achieved by multiple antennas at the base station.

2.3 LTE Physical Layer

Due to the huge different structures between eNodeB and User Equipment (UE), LTE PHY Downlink and Uplink are quite different. Therefore DL and UL are described separately in the
following sections. Because this thesis focuses on LTE Uplink structure, more details for UL at eNodeB side will be introduced.

2.3.1 Generic Frame Structure

There are two types of frame structure defined in the LTE specifications depending on the duplex schemes, type one is FDD and type two is TDD. The generic frame structure applies to both the LTE DL and UL.

![Figure 2.7 LTE generic frame structure shared by both UL and DL][9]

Figure 2.7 shows the generic frame structure of LTE. The duration for one radio frame is 10 msec. There are 20 slots in one frame numbered from 0 to 19. The duration for one slot is 0.5 msec. A sub-frame is defined as two consecutive slots. There are 10 sub-frames in one frame. There are 7 or 6 symbols in one slot depending on which kind of CP (normal or extended) is used. CP is inserted in front of every symbol.

2.3.2 Uplink

The LTE PHY specification is designed to accommodate bandwidths from 1.4 MHz to 20 MHz. Uplink multiplexing is accomplished via SC-FDMA. The basic sub-carrier spacing is 15 kHz. Table 2.3 summarizes SC-FDMA modulation parameters. The modulation schemes used in LTE uplink are BPSK, QPSK, 16QAM or 64QAM depending on the channel quality.
<table>
<thead>
<tr>
<th>Transmission BW</th>
<th>1.4 MHz</th>
<th>3 MHz</th>
<th>5 MHz</th>
<th>10 MHz</th>
<th>15 MHz</th>
<th>20 MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sub-frame duration</td>
<td>0.5 ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sub-carrier spacing</td>
<td>15 kHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sampling frequency</td>
<td>192MHz (1/2x3.84MHz)</td>
<td>3.84MHz</td>
<td>7.68MHz (2x3.84MHz)</td>
<td>15.36MHz (4x3.84MHz)</td>
<td>23.04MHz (6x3.84MHz)</td>
<td>30.72MHz (8x3.84MHz)</td>
</tr>
<tr>
<td>FFT size</td>
<td>128</td>
<td>256</td>
<td>512</td>
<td>1024</td>
<td>1536</td>
<td>2048</td>
</tr>
<tr>
<td>N_RB</td>
<td>6</td>
<td>15</td>
<td>25</td>
<td>50</td>
<td>75</td>
<td>100</td>
</tr>
<tr>
<td>Number of subcarriers</td>
<td>75</td>
<td>150</td>
<td>300</td>
<td>600</td>
<td>900</td>
<td>1200</td>
</tr>
<tr>
<td>SC-FDMA symbol per slot (short/long CP)</td>
<td>6/7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CP length (µsec/samples)</td>
<td>Short</td>
<td>(4.69/9)x6, (5.21/10)x1</td>
<td>(4.69/18)x6, (5.21/20)x1</td>
<td>(4.69/36)x6, (5.21/40)x1</td>
<td>(4.69/72)x6, (5.21/80)x1</td>
<td>(4.69/108)x6, (5.21/120)x1</td>
</tr>
</tbody>
</table>

Table 2.3 Uplink SC-FDE Modulation Parameters [10]

![LTE Physical Resource Blocks structure](image)

Figure 2.8 LTE Physical Resource Blocks structure
2.3.3 Multiplexing

Uplink physical resource blocks (PRBs) are assigned to UE by the base station (BS) scheduler via the downlink PDCCH (Physical Downlink Control CHannel). Uplink PRBs consist of 12 successive sub-carriers over a duration of one slot time. Figure 2.8 shows the basic structure of PRBs. Every symbol in the PRBs is called one resource element.

2.3.4 Physical Uplink Shared Channels

Physical channels are transmission channels carrying user data and control messages. Two types of UL physical channel are defined: Physical Uplink Shared Channel (PUSCH) and Physical uplink control channel PUCCH. This thesis will focus on PUSCH only. The main purpose for PUSCH is to transmit data. The modulation schemes are QPSK, 16QAM or 64QAM depending on the channel quality. Figure 2.9 shows the processing flow of PUSCH.

2.3.5 Uplink Reference Signal

Reference signals, also referred to as pilot signals which are previously known by both base station and UE, are used to estimate the channel condition. Two types of uplink reference signals are supported: Demodulation reference signal (DRS) and Sounding reference signal (SRS). Demodulation reference signal is assigned into the fourth SC-FDMA symbol of every slot and has the same size as the assigned resource. It is used to estimate the channel for data demodulation. Different from demodulation reference signal, the sounding reference signal is only used for scheduling. Both of them are based on Zadoff-Chu sequences.
Chapter 3

TD-LTE and FDD LTE

As described in chapter 2, two frame types are supported by LTE according to the duplex schemes (TDD and FDD) they are based on. LTE with TDD duplex scheme, also known as TD-LTE, is evolved from the existing TD-SCDMA technology operated by China Mobile. The main features of TD-LTE are asymmetric transmission data in UL/DL and unpaired spectrum. In this chapter, the main differences between TD-LTE and FDD LTE are discussed in the scope of baseband processing.

3.1 Frame Structure

The differences between TDD and FDD are mainly caused by their different frame structures (Figure 3.1). Both of them have 10 subframes for one radio frame with 10ms duration. But the frame structure for TDD is more complex than FDD. For one TDD radio frame there are two half frame and there are two special subframes in one radio frame. A special subframe consists of three fields: DwPTS (Downlink Pilot Time Slot), GP (Guard Period) and UpPTS (Uplink Pilot Time Slot). The subframes can be configured for different uplink/downlink requirements (figure 3.2).

When downlink subframe switch to uplink, a special subframe is needed between them for switching from downlink to uplink transmission. As table 3.1 shows, there are altogether 7 asymmetric UL/DL configurations, 0, 1, 2, 6 are 5ms DL-to-UL switch point period and 3, 4, 5 are 10ms DL-to-UL switch point period.

3.2 Features rooted from frame structures

The different frame structures of FDD and TDD lead to a series of changes, such as HARQ allocation, CQI/PMI feedback and synchronization signals. The main difference is in the Physical layer and it is not significant in the MAC, RLC or higher layer. The special subframe makes TD-LTE system has a number of features. Table 3.2 lists some new features derived from TD-LTE frame structure.
Figure 3.1 (a) Frame structure type1 FDD (b) Frame structure type2 TDD [9]

Table 3.1 Uplink-downlink configurations for TDD subframe [9]
(D / U stand for Downlink / Uplink subframe, S stands for special subframe used for a guard time)
### Table 3.2 Main features derived from TDD frame structure [13]

<table>
<thead>
<tr>
<th>Aspects</th>
<th>Differences</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Asymmetric UL/DL configuration</strong> – 7 configurations</td>
<td></td>
</tr>
<tr>
<td>SRS configuration</td>
<td>Different SRS opportunities for TDD</td>
</tr>
<tr>
<td>PRACH configuration</td>
<td>Different density and frequency/time position</td>
</tr>
<tr>
<td><strong>Special subframe design</strong> – [DwPTS + Gap + UpPTS]</td>
<td></td>
</tr>
<tr>
<td>SCH position</td>
<td>PSS and SSS position in TDD are different from FDD</td>
</tr>
<tr>
<td>Smaller Control region in DwPTS</td>
<td>2 OFDM symbols for control region in DwPTS</td>
</tr>
<tr>
<td>Punctured data transmission in DwPTS</td>
<td>PDSCH could be transmitted in DwPTS</td>
</tr>
<tr>
<td>SRS and PRACH in UpPTS</td>
<td>SRS in UpPTS can improve normal subframe PUSCH transmission</td>
</tr>
<tr>
<td>reciprocity since no PUCCH in UpPTS</td>
<td>SRS in UpPTS could be extended to larger bandwidth to exploit channel</td>
</tr>
<tr>
<td>Timing advance and additional offset</td>
<td>Gap accommodates the signal round trip time and DL-to-UL processing time</td>
</tr>
<tr>
<td></td>
<td>Additional offset accommodates the UL-to-DL processing time</td>
</tr>
</tbody>
</table>

### 3.3 Advantages and drawbacks

<table>
<thead>
<tr>
<th>Parameter</th>
<th>LTE-TDD</th>
<th>LTE-FDD</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Paired spectrum</strong></td>
<td>Supported</td>
<td>unsupported</td>
</tr>
<tr>
<td>Hardware cost</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>(no diplexer is needed to isolate the transmitter and receiver)</td>
<td>(Diplexer is needed and cost is higher for the UEs)</td>
<td></td>
</tr>
<tr>
<td>UL/DL asymmetry</td>
<td>Dynamic configurable</td>
<td>Fixed by frequency allocation.</td>
</tr>
<tr>
<td>Guard period / guard band</td>
<td>Guard period is required to ensure uplink and downlink transmissions do not clash. (Large guard period will limit capacity.)</td>
<td>Guard band is required to provide sufficient isolation between uplink and downlink. (Large guard band does not impact capacity.)</td>
</tr>
<tr>
<td>Discontinuous transmission</td>
<td>Discontinuous transmission</td>
<td>Continuous transmission</td>
</tr>
<tr>
<td>(This can degrade the performance of the RF power amplifier in the transmitter.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cross slot interference</td>
<td>BS needs to be synchronized to the UL and DL transmission times respectively. If neighboring BSs use different UL and DL assignments and share the same channel, interference may occur between cells.</td>
<td>Not applicable</td>
</tr>
<tr>
<td>mobility</td>
<td>120km/h at the most</td>
<td>500km/h at the most</td>
</tr>
</tbody>
</table>

**Table 3.3 Advantages / disadvantages of LTE TDD and LTE FDD**
All in all, the difference between TDD and FDD is to a large extent only in the frame structure. At the technical level, in order to maintain a high consistency with FDD, TDD uses the same technology including multiple access methods (OFDMA for DL, SC-FDMA for UL), multi-antenna transmission and so on. Thus the advantages of TD-LTE will be more concentrated in a limited spectrum usage and to the use of channel reciprocity technology.
Chapter 4

Computational Complexity Analysis

4.1 Overall System Flow

Figure 4.1 and 4.3 illustrate LTE uplink and downlink system model. As mentioned previously, this thesis will focus on computational intensive part on eNodeB side from the flows below, such as FFT/IFFT, channel estimation, equalization. Brief explanations for every stage will be given first and then complexity analysis will be carried out. Later in this thesis, cost analysis will be based on functions shown in Figure 4.3 at a typical scenario (20MHz bandwidth, slow-fading channel) for Uplink at eNodeB side.

![LTE Downlink System Model - OFDMA](image)

Figure 4.1 Downlink system model for LTE [9] [15]
4.1.1 LTE Downlink

Suppose the raw binary bits are ready to transmit from eNodeB to UEs. Downlink signal is produced through several stages described below [9] [15].

Transport block CRC attachment: CRC bits are calculated and attach to the initial raw bits.

Code block segmentation & Code block CRC attachment: This stage is to divide the bits into blocks. The block size $Z=6144$ and every blocks should perform additional CRC attachment. After the processing, the blocks are going to perform channel coding.

Turbo coding: For every block, turbo coding is performed. The scheme of turbo encoder is a Parallel Concatenated Convolutional Code (PCCC) with two 8-state constituent encoders and one internal interleaver for scatting error burst. The coding rate is 1/3. Turbo coding provides error correction function.

Interleaving: The three output bit streams derived from turbo coding are interleaved separately. The purpose for this process is to avoid burst errors.

Rate matching: Rate matching is to match the block size to the radio frame by repeating bits to increase the rate or puncturing bits to decrease the rate.

Code block concatenation: This stage is to concatenate the coded blocks.

Scrambling: The block of bits is scrambled with a UE-specific scrambling sequence prior to modulation [9]. The main reason for scrambling here is to making the transmitted data more dispersed to meet maximum power spectral density requirements [16].

Modulation mapping: This stage is to map the binary bits into complex value symbols by using QPSK, 16QAM and 64QAM modulation schemes, corresponding to two, four and six bits per modulation symbol. Which modulation scheme will be used is determined by the channel quality and the requirements of data rats for transmission.

Layer mapping: For each code word, the complex-valued modulation symbols will be mapped onto one, two, three or four layers. Two kinds of layer mapping are supported in LTE for spatial multiplexing and for transmit diversity respectively.

Precoding: Precoding is performed to map the complex-valued modulation symbols from the layers to multiple antennas. Precoding has two schemes according to different layer mapping methods. Layer mapping and precoding are also known as antenna mapping.
Pilot Insertion: Pilot symbols are generated and inserted to complex-valued modulation symbols on each antenna port. Figure 4.2 shows the structure for LTE downlink pilot symbol. The positions for pilot symbols of one antenna port are not used at other antenna port.

Figure 4.2 LTE downlink pilot symbol structure [9]

Resource element mapping: This stage is to map the complex-valued modulation symbols to the physical resource blocks at every antenna port. The mapping shall be in increasing order of first
resource block index $k$ over the assigned physical resource blocks and then the index $l$, starting with the first slot in one subframe [9].

IFFT: $N$-point IFFTs are performed to convert the signal from frequency domain to time domain after the resource element mapping starting from symbol index $l=0$. The size of $N$ is listed in Table 4.1.

Add CP & PS: Attach CP into every symbol and then perform PS. The CP length is defined in [9].

DAC & RF: Convert digital signal to analog signal and then transmit from the radio frequency.

**LTE UPLINK SYSTEM MODEL - SC-FDMA**

---

**Figure 4.3** Uplink system model for LTE [9] [15]
4.1.2 LTE Uplink

Although SC-FDMA is the multiple access schemes for LTE uplink, most baseband signal processing methods are similar.

RF & ADC: eNodeB receive analog signal from RF and then convert to digital signal.

SP & Remove CP: Perform SP and then remove CP.

FFT: N-point FFTs are performed to convert the signal from time domain to frequency domain. The size of N is listed on Table 4.1.

User Extraction: Extract every user’s symbol data on different subcarriers according to their PRBs configurations.

Channel Estimation: Based on the pilot symbols extracted from the frame, estimate channel matrix $H$. Since this is a computational intensive part at the baseband, detailed discussion with complexity analysis is followed in the later section.

Equalization: Based on the estimated channel matrix $H$, perform equalization on the whole slot.

MIMO combination: If multiple antennas are involved, the received signal from different antennas needs to be combined according to the MIMO scheme implemented.

Remove Pilot: Remove pilot symbol from the modulation symbol frame.

Resource element demapping: Demapping the complex-valued modulation symbol frame into blocks.

IFFT: M-point IFFTs are performed to convert the data from frequency domain to time domain. Here the size of M is not power-of-2, so the radix-2 FFT algorithm is not applicable.

Soft demapping: Convert the received SC-FDMA symbols into soft bits according to the modulation scheme employed.

De-scrambling: This is the inverse stage of scrambling.

Channel De-interleaver: De-interleaver for rank indication bits, HARQ-ACK information bits and PUSCH/CQI multiplexing bits.

Data and control demultiplexing: Demultiplexing both PUSCH data and CQI bits.

Code block deconcatenation: This stage is to segment the received bits into blocks.
Rate dematching: For every code blocks, rate dematching makes the code bits into three streams.

Turbo decoding: Turbo decoder is built in the similar way as the encoder. It uses soft decision to give the code block bits.

Code block CRC Removal: Perform CRC check and then remove 24 parity bits in each code blocks.

Code block de-segmentation: Combine all the code blocks and get the binary bits with parity bits.

Transport block CRC Removal: Perform CRC check and then remove 24 parity bits.

4.2 Complexity Analysis for LTE supported FFT/IFFT

Since FFT and IFFT are implemented both in uplink and downlink, in the uplink UE has an M-point DFT transform precoding while eNodeB will also do an M-point IDFT after user extraction stage. Table 4.1 lists the supported N-point FFT size for LTE downlink and uplink with different bandwidth configurations. All of them except 1536 point FFT at 15MHz are power-of-2 based FFT which can be computed using the radix-2 FFT algorithm. For SC-FDMA based uplink model,

\[ M_{RB}^{PUSCH} = M_{RB}^{PUSCH} \cdot N_{RB}^{sc} \]

\[ M_{RB}^{PUSCH} \] must be multiple of 2, 3 or 5 [9]. Table 4.2 lists the possible values of \( M_{sc}^{PUSCH} \). Due to the size of DFT is not the power-of-2, traditional radix-2 FFT algorithm is not applicable. To solve the problem, a divided and conquer mixed-radix FFT algorithm is introduced. We will start the complexity analysis by studying the basic FFT algorithm first.

<table>
<thead>
<tr>
<th>Transmission BW</th>
<th>1.4 MHz</th>
<th>3 MHz</th>
<th>5 MHz</th>
<th>10 MHz</th>
<th>15 MHz</th>
<th>20 MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFT size</td>
<td>128</td>
<td>256</td>
<td>512</td>
<td>1024</td>
<td>1536</td>
<td>2048</td>
</tr>
</tbody>
</table>

**Table 4.1 LTE downlink/uplink N-point FFT size**

<table>
<thead>
<tr>
<th>( M_{RB}^{PUSCH} )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>12</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M_{RB}^{PUSCH} )</td>
<td>12</td>
<td>24</td>
<td>36</td>
<td>48</td>
<td>60</td>
<td>72</td>
<td>96</td>
<td>108</td>
<td>120</td>
<td>144</td>
<td>180</td>
<td>192</td>
</tr>
<tr>
<td>( M_{RB}^{PUSCH} )</td>
<td>18</td>
<td>20</td>
<td>24</td>
<td>25</td>
<td>27</td>
<td>30</td>
<td>32</td>
<td>36</td>
<td>40</td>
<td>45</td>
<td>48</td>
<td>50</td>
</tr>
<tr>
<td>( M_{RB}^{PUSCH} )</td>
<td>216</td>
<td>240</td>
<td>288</td>
<td>300</td>
<td>324</td>
<td>360</td>
<td>384</td>
<td>432</td>
<td>480</td>
<td>540</td>
<td>576</td>
<td>600</td>
</tr>
<tr>
<td>( M_{RB}^{PUSCH} )</td>
<td>54</td>
<td>60</td>
<td>64</td>
<td>72</td>
<td>75</td>
<td>80</td>
<td>81</td>
<td>90</td>
<td>96</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( M_{RB}^{PUSCH} )</td>
<td>648</td>
<td>720</td>
<td>768</td>
<td>864</td>
<td>900</td>
<td>960</td>
<td>972</td>
<td>1080</td>
<td>1152</td>
<td>1200</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 4.2 LTE uplink Transform Precoding M-point DFT size**
4.2.1 Fast Fourier transform and Inverse FFT

A fast Fourier transform is the algorithm to calculate the discrete Fourier transform (DFT) quickly and efficiently. It is widely used in many fields, especially in digital signal processing filed. Let \( x_0, \ldots, x_{N-1} \) be complex numbers. The definitions of DFT are as follows:

\[
X_k = \sum_{n=0}^{N-1} x_n e^{-j\frac{2\pi nk}{N}} = \sum_{n=0}^{N-1} x_n W_N^{nk} \quad k = 0, \ldots, N - 1.
\]  

(4.1)

The computation complexity of DFT is \( O(N^2) \), it needs \( N^2 \) complex multiplication and \( N^2 \) complex additions.

Cooley-Tukey algorithm is the most popular FFT algorithm and was proposed by J.W.Cooley and J.W.Tukey in 1965. It is based on a divide and conquer algorithm that recursively divide a DFT into many smaller DFTs [17]. Such FFT algorithm can reduced the complexity of DFT to \( O(N \log N) \). It has two functionally equivalent forms known as decimation in time (DIT) and decimation in frequency (DIF). Both forms have the same computation complexity. Radix-2 and Radix-4 is the most common FFT algorithms.

Since inverse FFT can be calculated by implementing FFT algorithm, their computational complexities are at same level. The method for computing IFFT is for an N-point complex data: firstly, change the real and imagine part of the data, then compute the FFT on the data, lastly use \( 1/N \) multiply the FFTed data, the result data is just the IFFT result for the N-point complex data.

4.2.2 Radix-2 FFT

Radix-2 FFT algorithms are the simplest FFT algorithms. Two methods can be used for calculating based on radix-2 FFT algorithms, namely Decimation in Time (DIT) and Decimation in Frequency (DIF). The mainly difference is that for DIT algorithm the input signal must do an bit-reverse implementation first whereas for DIF algorithm the output signal must do an bit-reverse implementation at last. Figure 4.4 shows the similar butterfly computing schemes of DIT and DIF.

Consider a DFT of \( N=2^n \) points [19], divide the N points data into two sets of \( N/2 \) points data, \( g_1(n) \) and \( g_2(n) \), respectively.

\[
g_1(n) = x_{2n},
g_2(n) = x_{2n+1}, \quad n = 0, 1, \ldots, N / 2 - 1
\]

(4.2)
Now rewrite the formula of the DFT:

\[ X_k = \sum_{n=0}^{N-1} x_n e^{\frac{2\pi i nk}{N}} \quad k = 0, \ldots, N - 1. \]

\[ = \sum_{n \text{ even}} x_n W_N^{kn} + \sum_{n \text{ odd}} x_n W_N^{kn} \]

\[ = \sum_{m=0}^{N/2-1} x_{2m} W_N^{2km} + \sum_{m=0}^{N/2-1} x_{2m+1} W_N^{2(2m+1)} \]

\[ = \sum_{m=0}^{N/2-1} g_1(m) W_N^{2km} + \sum_{m=0}^{N/2-1} g_2(m) W_N^{2k(m+1/2)} \]

\[ = G_1(k) + W_N^k G_2(k) \quad k = 0, 1, \ldots, N - 1 \]

Because \( G_1(k) \) and \( G_2(k) \) are periodic,

\[ G_1(k) = G_1(k + \frac{N}{2}) , \quad G_2(k) = G_2(k + \frac{N}{2}) \] (4.4)

and \( W_N^{k+N/2} = -W_N^k \), \( (W_N^{N/2})^2 = -1 \), \( X_k \) can be expressed by:

\[ X_k = G_1(k) + W_N^k G_2(k) \quad , \quad k = 0, 1, \ldots, \frac{N}{2} \]

\[ X_{k+N/2} = G_1(k) - W_N^k G_2(k) \quad , \quad k = 0, 1, \ldots, \frac{N}{2} \] (4.5)

Rewrite it into matrix form:

\[
\begin{bmatrix}
X_k \\
X_{k+N/2}
\end{bmatrix} = \begin{bmatrix}
1 & 1 \\
1 & -1
\end{bmatrix} \begin{bmatrix}
W_N^0 G_1(k) \\
W_N^k G_2(k)
\end{bmatrix}
\] (4.6)
Note $G_1(k)$ and $G_2(k)$ are two $N/2$ points DFTs of the data sets $g_1(n)$ and $g_2(n)$ respectively. The computation complexity for two $N/2$ points DFTs need about $N^2/2$ complex multiplications and complex additions. The complexity reduces to nearly 50% by using this method recursively to calculate $G_1(k)$ and $G_2(k)$, only 2 points DFT need to be computed in the end.

For an $N=2^m$ points radix-2 FFT, there are $\log_2N=m$ stages and every stage has $N/2$ butterflies, so the total computation complexity for radix-2 FFT should be $0.5N(\log_2 N)$ complex multiplications and $N(\log_2 N)$ complex additions. Figure 4.5 shows an 8-point radix-2 DIT FFT algorithm.

![8-point radix-2 DIT FFT algorithm](image)

Furthermore, the nontrivial complexity for radix-2 FFT is $0.5N\log_2N-N+1$ complex multiplication and $N\log_2N$ complex additions by ignoring the twiddle factors with the power of 0.

### 4.2.3 Radix-4 FFT

Consider $N=4^m$ points DFT, similarly to radix-2 FFT algorithm, divide the N point data into 4 sets of $N/4$ points data. The definition of N points DFT can be rewrite [19]:

- **Stage 1**: Divide the data into 4 sets of $N/4$ points data. Calculate the DFT of each set.
- **Stage 2**: Combine the results from Stage 1 to form $N/2$ points DFTs.
- **Stage 3**: Recursively apply the same process to obtain the final $N$ points DFT.
Instead of directly computing N points DFT, the result can be derived from computing 4 sets of 
$N/4$ point DFTs. To make it clearly, rewrite above formula in matrix form:

\[
\begin{align*}
X(p,q) &= \sum_{l=0}^{N/4-1} W_{N/4}^{lp} F(l,q) W_4^{-lp} \\
F(l,q) &= \sum_{m=0}^{N/4-1} x(l,m) W_{N/4}^{-mq} \\
p, l = 0,1,2,3; \quad q = 0,1,2,\ldots, \frac{N}{4} - 1 \\
\end{align*}
\]  

(4.7)

and

\[x(l,m) = x(4m + l)\]

\[X(p,q) = X\left(\frac{N}{4} p + q\right)\]

So it employs three complex multiplications ($W_{N}^{0}$ = 1 ) and 12 complex additions. By 
decomposing the twiddle factor matrix, it is possible to reduce the complex additions. Here is the 
decomposing algorithm [18] (Figure 4.7):

![Figure 4.6 Radix-4 DIT FFT butterfly computation structure [19]](image)

So it employs three complex multiplications ($W_{N}^{0}$ = 1 ) and 12 complex additions. By 
decomposing the twiddle factor matrix, it is possible to reduce the complex additions. Here is the 
decomposing algorithm [18] (Figure 4.7):
Figure 4.7 A recursive decomposing method for DFT calculation [18]

Using this algorithm, the twiddle factor matrix can be rewrite,
The matrix form is as follows now [19]:

\[
\begin{bmatrix}
X(0,q) \\
X(1,q) \\
X(2,q) \\
X(3,q)
\end{bmatrix} =
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 1 & 0 \\
0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
1 & 0 & -1 & 0 \\
0 & 1 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
W_N^q F(0,q) \\
W_N^q F(1,q) \\
W_N^q F(2,q) \\
W_N^q F(3,q)
\end{bmatrix}
\]  
(4.10)

This butterfly needs three complex multiplications and only eight (4+4) complex additions.

Figure 4.8 16-point Radix-4 DIT FFT algorithm (normal input and bit-reversed output) [19]
A 16-point radix-4 DIT FFT algorithm is shown in Figure 4.8. For an \(N=4^m\) points radix-4 FFT, there are \(\log_4 N = m\) stages and every stage has \(N/4\) butterflies, so the total computation complexity for radix-4 FFT should be \(3N \log_4 N = \frac{3N}{8} \log_2 N\) complex multiplications and \(2N \log_4 N = N \log_2 N\) complex additions.

### 4.1.4 Split-Radix FFT

Split-radix FFT algorithm, first introduced by R. Yavne in 1968 [20], is the most efficient power-of-two FFT algorithms so far. It mixes radix-2 and radix-4 decompositions, achieves about two-third multiplications than the radix-2 needs and the same additions complexity. It is proved that split-radix FFT algorithm has lower complexity than radix-2, radix-4 or any other higher-radix power-of-two FFT [21].

Unfortunately, although the irregular butterfly structure brings reduced computational complexity, the increased programming complexity makes it hard to implement on hardware. In other words, it may be difficult to code split-radix FFT algorithm for vector or multi-core computers.

### 4.2.5 Radix-3, Radix-5 and Radix-\(r\) FFT

Consider \(N=3^m\) points DFT, it can be rewrote similarly to Radix-4 FFT algorithm:

\[
X(p, q) = \sum_{l=0}^{2} [W_N^{pl} F(l, q)] W_N^{ql} \\
F(l, q) = \sum_{m=0}^{N/3-1} x(l, m) W_N^{mq} \\
p, l = 0, 1, 2 ; \quad q = 0, 1, 2, ..., \frac{N}{3} - 1
\]
and

\[ x(l, m) = x(3m + l) \]

\[ X(p, q) = X \left( \frac{N}{3} p + q \right) \]  \hspace{1cm} (4.11)

Rewrite the formula into matrix form:

\[
\begin{bmatrix}
X(0, q) \\
X(1, q) \\
X(2, q)
\end{bmatrix} =
\begin{bmatrix}
1 & 1 & 1 \\
W_3^1 & W_3^2 & W_3^3 \\
W_3^2 & W_3^4 & W_3^{10}
\end{bmatrix}
\begin{bmatrix}
W_N^0 F(0, q) \\
W_N^1 F(1, q) \\
W_N^2 F(2, q)
\end{bmatrix} =
\begin{bmatrix}
1 & 1 & 1 \\
W_3^1 & W_3^2 & W_3^3 \\
W_3^2 & W_3^4 & W_3^{10}
\end{bmatrix}
\begin{bmatrix}
W_N^0 F(0, q) \\
W_N^1 F(1, q) \\
W_N^2 F(2, q)
\end{bmatrix}
\]  \hspace{1cm} (4.12)

\[ W_3^1 = \exp(-2\pi j / 3), W_3^2 = \exp(-4\pi j / 3) \]

Because \( W_3^1 \) and \( W_3^2 \) are complex number, it needs 4 plus 2 altogether 6 complex multiplications and six complex additions. Here the twiddle factor matrix cannot be decomposed as it does in radix-4 algorithm and there are complex numbers in it. Both of them make it inefficient compared to radix-2 and radix-4 algorithms. The total computation complexity is \( 2N \log_3 N \) complex multiplications and \( 2N \log_3 N \) complex additions. Similarly for radix-5 FFT, the total computational complexity is \( 4N \log_5 N \) complex multiplications and complex additions. Generally speaking, for an \( N=r^m \) (r is prime number) points DFT, the total computational complexity is \( (r-1)N \log_r N \) complex multiplications and complex additions. Table 4.3 and Table 4.4 shows the complexity analysis result for N-point radix-r FFT (where 1 complex multiplication equals to 4 real multiplications plus 2 real additions and 1 complex addition equals to 2 real additions).

Moreover, the nontrivial complexity for radix-r (r is prime number) N-point FFT is \( (r-1)N \log_r N - N+1 \) complex multiplication and \( (r-1)N \log_r N \) complex additions by ignoring the twiddle factors with the power of 0.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.5N \log_2 N</td>
<td>2N \log_2 N</td>
<td>N \log_2 N</td>
<td>3N \log_2 N</td>
</tr>
<tr>
<td>3</td>
<td>2N \log_3 N</td>
<td>8N \log_3 N</td>
<td>2N \log_3 N</td>
<td>8N \log_3 N</td>
</tr>
<tr>
<td>4</td>
<td>0.375N \log_2 N</td>
<td>1.5N \log_2 N</td>
<td>N \log_2 N</td>
<td>1.75N \log_2 N</td>
</tr>
<tr>
<td>5</td>
<td>4N \log_5 N</td>
<td>16N \log_5 N</td>
<td>4N \log_5 N</td>
<td>16N \log_5 N</td>
</tr>
<tr>
<td>r (r is prime)</td>
<td>(r-1)N \log_r N</td>
<td>4(r-1)N \log_r N</td>
<td>(r-1)N \log_r N</td>
<td>4(r-1)N \log_r N</td>
</tr>
</tbody>
</table>

**Table 4.3** Radix-r FFT complexity
### Table 4.4 N-point Radix-\( r \) FFT complexity

<table>
<thead>
<tr>
<th>( N )</th>
<th>( \text{Radix-2} )</th>
<th>( \text{Radix-3} )</th>
<th>( \text{Radix-4} )</th>
<th>( \text{Real Multiplications} )</th>
<th>( \text{Real Additions} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>48</td>
<td></td>
<td></td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>144</td>
<td></td>
<td></td>
<td>108</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>128</td>
<td>96</td>
<td>192</td>
<td>112</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>648</td>
<td></td>
<td></td>
<td>486</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>320</td>
<td></td>
<td></td>
<td>480</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>768</td>
<td>576</td>
<td>1152</td>
<td>672</td>
<td></td>
</tr>
<tr>
<td>81</td>
<td></td>
<td>2592</td>
<td></td>
<td>1944</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>1792</td>
<td></td>
<td></td>
<td>2688</td>
<td></td>
</tr>
<tr>
<td>243</td>
<td>9720</td>
<td></td>
<td></td>
<td>7290</td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>4096</td>
<td>3072</td>
<td>6144</td>
<td>3584</td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>9216</td>
<td></td>
<td>13824</td>
<td></td>
<td></td>
</tr>
<tr>
<td>729</td>
<td>34992</td>
<td>26244</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1024</td>
<td>20480</td>
<td>15360</td>
<td>30720</td>
<td>17920</td>
<td></td>
</tr>
<tr>
<td>2048</td>
<td>45056</td>
<td></td>
<td>67584</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2187</td>
<td>122472</td>
<td></td>
<td>91854</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### 4.2.6 Mixed-radix Divided and Conquer DFT complexity

Similar as the decomposing method introduced in radix-2 and radix-4 sections, a Divided and Conquer strategy [22] can be used to divide the mixed-radix DFT into small parts, recursively compute every part and then combine the results.

Suppose \( N = L \times M \) for equation (4.1), it can be expressed by using a 2D mapping:

Input: \( n = l + mL, 0 \leq l \leq L, 0 \leq m \leq M \)

Output: \( k = Mp + q, 0 \leq p \leq L, 0 \leq q \leq M \)

With this mapping, the \( N \) point DFT can be split to two smaller \( L \) point and \( M \) point DFTs:

\[
X(p, q) = \sum_{m=0}^{M-1} \sum_{l=0}^{L-1} x(l, m) e^{-j2\pi(Mp+q)(mL+l)/N} \\
= \sum_{l=0}^{L-1} \left( e^{j2\pi q/L} \sum_{m=0}^{M-1} x(l, m) e^{-j2\pi mq/M} \right) e^{-j2\pi lq/L} (4.13)
\]
Using this method, suppose \( M = 2^{k_1} \cdot 3^{k_2} \cdot 5^{k_3} \)

Define \( M = \prod_{i=0}^{n} s_i \), \( s_0 = 2^{k_1} \), \( s_1 \sim s_{k_2} = 3 \), \( s_{k_2+1} \sim s_{k_2+k_3} = 5 \)

The number of nontrivial real operations for calculating 5-point FFT is denoted by:

\[ C_{5-FFT} = 20 \text{ complex additions} + 16 \text{ complex multiplication} = 40 + 96 = 136 \text{ real operations} \]

The number of nontrivial real operations for calculating 3-point FFT is denoted by:

\[ C_{3-FFT} = 6 \text{ complex additions} + 4 \text{ complex multiplication} = 12 + 24 = 36 \text{ real operations} \]

The number of nontrivial real operations for calculating the radix-2 \( s_0 \) point FFT is denoted by:

\[ C_{\text{nontrivial-radix-2}} = 0.5N \log_2 N \cdot N + 1 \text{ complex multiplication} + N \log_2 N \text{ complex addition} = 5N \log_2 N \cdot 6N + 6 \text{ real operations} \]

The complexity of \( M \)-point DFT using D&C algorithm should be:

\[
C_{D&C} = \left( \prod_{i=0}^{n-1} s_i \right) \cdot C_{5-FFT} + \left( \prod_{i=0}^{n-2} s_i \right) \cdot (s_{n-1} - 1)(s_n - 1) \cdot C_{\text{Complex Multiplication}}

+ \left( \prod_{i=0}^{n-3} s_i \right) \cdot (s_{n-2} - 1)(s_n \cdot s_{n-1} - 1) \cdot C_{\text{Complex Multiplication}}

+ \cdots

+ \left( \prod_{i=0}^{n-k_3} s_i \right) \cdot \left( \prod_{i=n-k_3+1}^{n} s_i \right) \cdot C_{5-FFT} + \left( \prod_{i=n-k_3+2}^{n} s_i \right) \cdot (s_{n-k_3} - 1) \cdot \left( \prod_{i=n-k_3+1}^{n} s_i - 1 \right) \cdot C_{\text{Complex Multiplication}}

+ \cdots

+ \left( \prod_{i=0}^{n} s_i \right) \cdot (s_{n-1} - 1)(\prod_{i=3}^{n} s_i - 1) \cdot C_{\text{Complex Multiplication}}

+ s_0 \cdot (\prod_{i=3}^{n} s_i - 1) \cdot C_{\text{Complex Multiplication}}

+ (\prod_{i=1}^{n} s_i) \cdot (5s_0 \cdot \log_2 (s_0) - 6s_0 + 6)
\[ M \cdot C_{5-FFT} / s_n + \left( \prod_{i=0}^{n-3} s_i \right) \cdot (s_{n-1} - 1)(s_n - 1) \cdot C_{\text{Complex Multiplication}} \]
\[ + M \cdot C_{5-FFT} / s_{n-1} + \left( \prod_{i=0}^{n-3} s_i \right) \cdot (s_{n-2} - 1)(s_n \cdot s_{n-1} - 1) \cdot C_{\text{Complex Multiplication}} \]
\[ + \ldots \]
\[ + M \cdot C_{5-FFT} / s_{n-k3+1} + \left( \prod_{i=0}^{n-k3-1} s_i \right) \cdot (s_{n-k3} - 1)(\prod_{i=n-k3+1}^{n} s_i - 1) \cdot C_{\text{Complex Multiplication}} \]
\[ + M \cdot C_{3-FFT} / s_{n-k3} + \left( \prod_{i=0}^{n-k3-1} s_i \right) \cdot (s_{n-k3-1} - 1)(\prod_{i=n-k3}^{n} s_i - 1) \cdot C_{\text{Complex Multiplication}} \]
\[ + \ldots \]
\[ + M \cdot C_{3-FFT} / s_2 + s_0 \cdot (s_1 - 1)(\prod_{i=2}^{n} s_i - 1) \cdot C_{\text{Complex Multiplication}} \]
\[ + M \cdot C_{3-FFT} / s_1 + (s_0 - 1)(\prod_{i=2}^{n} s_i - 1) \cdot C_{\text{Complex Multiplication}} \]
\[ + 5M \cdot \log_2(s_0) - 6M + 6M / s_0 \]
\[ = M \cdot C_{5-FFT} \cdot \sum_{i=n-k3+1}^{n} \frac{1}{s_i} + M \cdot C_{3-FFT} \cdot \sum_{i=1}^{n-k3} \frac{1}{s_i} \]
\[ + 5M \cdot \log_2(s_0) - 6M + 6M / s_0 \]
\[ + (nM + 1 - \sum_{i=0}^{n} \frac{M}{s_i}) \cdot 6 \]
\[ = M \cdot (136 \cdot k3 / 5 + M \cdot 36 \cdot k2 / 3 + 5M \cdot k1 - 6M + 6M / s_0 + (nM + 1 - \sum_{i=0}^{n} \frac{M}{s_i}) \cdot 6 \]
\[ = M \cdot (5k1 + 16k2 + 32k3 - 6) + 6 \]

(4.14)

For example, the complexity for a 12 point Mix-radix FFT (Figure 4.10) should be that of four 3-point DFT, three 4-point raidx-2 FFT plus the twiddle factor multiplications: 4*36+3*22+6*6=246 real operations.
According to [9], the number of the RBs assigned to the UEs is defined as the multiple of 2, 3, or 5. For one resource block, there are $N_{Ul}^{SC} = 12$ subcarriers. All the computational complexity of the DFT with possible size is shown in Table 4.4 from index 1 to 34. The complexity for 1536-point DFT is also listed here because it is used for 15MHz bandwidth though this DFT is not in the same scope with the other 34 DFTs listed in the table.
<table>
<thead>
<tr>
<th>Index</th>
<th>DFT Size</th>
<th>k1</th>
<th>k2</th>
<th>k3</th>
<th>ROs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>246</td>
</tr>
<tr>
<td>2</td>
<td>24</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>606</td>
</tr>
<tr>
<td>3</td>
<td>36</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1302</td>
</tr>
<tr>
<td>4</td>
<td>48</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>1446</td>
</tr>
<tr>
<td>5</td>
<td>60</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>3126</td>
</tr>
<tr>
<td>6</td>
<td>72</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>2958</td>
</tr>
<tr>
<td>7</td>
<td>96</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>3366</td>
</tr>
<tr>
<td>8</td>
<td>108</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>5622</td>
</tr>
<tr>
<td>9</td>
<td>120</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>6846</td>
</tr>
<tr>
<td>10</td>
<td>144</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>6630</td>
</tr>
<tr>
<td>11</td>
<td>180</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>12246</td>
</tr>
<tr>
<td>12</td>
<td>192</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>7686</td>
</tr>
<tr>
<td>13</td>
<td>216</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>12318</td>
</tr>
<tr>
<td>14</td>
<td>240</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>14886</td>
</tr>
<tr>
<td>15</td>
<td>288</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>14694</td>
</tr>
<tr>
<td>16</td>
<td>300</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>25206</td>
</tr>
<tr>
<td>17</td>
<td>324</td>
<td>2</td>
<td>4</td>
<td>0</td>
<td>22038</td>
</tr>
<tr>
<td>18</td>
<td>360</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>26286</td>
</tr>
<tr>
<td>19</td>
<td>384</td>
<td>7</td>
<td>1</td>
<td>0</td>
<td>17286</td>
</tr>
<tr>
<td>20</td>
<td>432</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>26790</td>
</tr>
<tr>
<td>21</td>
<td>480</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>32166</td>
</tr>
<tr>
<td>22</td>
<td>540</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>45366</td>
</tr>
<tr>
<td>23</td>
<td>576</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td>32262</td>
</tr>
<tr>
<td>24</td>
<td>600</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>53406</td>
</tr>
<tr>
<td>25</td>
<td>648</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>47310</td>
</tr>
<tr>
<td>26</td>
<td>720</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>56166</td>
</tr>
<tr>
<td>27</td>
<td>768</td>
<td>8</td>
<td>1</td>
<td>0</td>
<td>38406</td>
</tr>
<tr>
<td>28</td>
<td>864</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>57894</td>
</tr>
<tr>
<td>29</td>
<td>900</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>90006</td>
</tr>
<tr>
<td>30</td>
<td>960</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>69126</td>
</tr>
<tr>
<td>31</td>
<td>972</td>
<td>2</td>
<td>5</td>
<td>0</td>
<td>81654</td>
</tr>
<tr>
<td>32</td>
<td>1080</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>96126</td>
</tr>
<tr>
<td>33</td>
<td>1152</td>
<td>7</td>
<td>2</td>
<td>0</td>
<td>70278</td>
</tr>
<tr>
<td>34</td>
<td>1200</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>112806</td>
</tr>
<tr>
<td>35</td>
<td>1536</td>
<td>9</td>
<td>1</td>
<td>0</td>
<td>84486</td>
</tr>
</tbody>
</table>

Table 4.5 Mixed-radix Divided and Conquer DFT complexity in MACs and Flops
4.3 Synchronization for Uplink

In LTE uplink, an UE must be synchronized to the BS first in order to transmit data to BS. This process is initialized in Physical Random Access Channel (PRACH). Both the receiver and transmitter chains of the UE are driven by the same clock which is assumed to be locked to the downlink channel. The accuracy is assumed to be 0.1 ppm [23]. The timing and frequency error (carrier frequency offset) as a whole will result in the time drift in both the downlink and uplink channel. Based on the assumption of the phase error, the time drift is around 0.1 μs per second. The time drift in uplink is compensated by automatically when it is compensated in the downlink. Hence only the synchronization on the downlink is needed in respective with the clock drift. However, the time drifts resulting from the Doppler Effect will have to be estimated by the eNodeB based on the uplink transmission. This is done using the 800μs LTE PRACH sequence which is built from cyclic-shifting a ZC sequence.

4.3.0 Cell Search Procedure

Whenever a UE is switched on or when it has lost the connection to the serving cell, it will search for a cell and get the information of downlink scrambling code and frame synchronization of that cell. This is also called initial synchronization. The cell search procedure consists of three steps.

The first step is Slot synchronization. During this step, the UE achieves slot synchronization with the cell by the help of Primary Synchronization Signal (PSS). The second step is to perform frame synchronization and identify the code-group of the cell found in the first step by analyzing the Secondary Synchronization Signal (SSS). Time synchronization is completed at the end of step 2. The last step called Scrambling-code identification is to identify the exact primary scrambling code used by the cell found in the previous step [24]. Frequency synchronization is also performed in the UE by analyzing the received data.

4.3.1 Random Access Procedure

The random access procedure includes the following steps.

1. [eNodeB] Cells broadcast the cells information through PBCH to all the UEs within the cells.
2. [UE] UEs random select the available preamble signatures and the access time of the current cell according to the information derived from Acquisition Indication Channel (AICH).
3. [UE] UEs determine the initial transmit power according to the pilot signal.
4. [UE] UEs start to transmit the preamble signature at the initial transmit power at the specified access time.
5. [eNodeB] Cells receive the random access request sent from UEs and feedback Random Access Response (RAR) to UEs with the UE identity. If a contention is detected, which
means several UEs are using the same preamble signatures, BS will feedback RAR to all these UEs with the successfully detected UE identity.

(6) [UE] UEs monitor the RAR information from the BS through AICH.

(7) [UE] The UE correctly decodes the RAR message and detects other UE’s identity, which means a contention is occurred, and then sends nothing back to the cell; The UE fails to decode the RAR message and sends nothing back to the cell; The UE doesn’t receive the RAR message in a specified time.

(8) [UE] If (7) occurs, the UE will wait for a moment, reselect the preamble signature and the access time from AICH, and retransmit the preamble signature at a higher transmit power.

(9) [UE] If the maximum retransmit limit is reached, the UE will give up which means the random access procedure failed.

(10) [UE] The UE correctly decodes the RAR message and detects its own identity, which means the random access procedure is succeed, and then sends back a positive ACKnowledgement (ACK) to the cell, which means random access procedure succeed.

4.3.2 Preamble sequence

Five preamble formats are defined in LTE [9], format 0-3 are supported in both FDD and TDD schemes. Format 4 is designed to fit into UpPTS of the special subframe and thus supported for TDD only (Table 4.6).

<table>
<thead>
<tr>
<th>Preamble format</th>
<th>$T_{CP}$</th>
<th>$T_{SEQ}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3168 • $T_S$</td>
<td>24576 • $T_S$</td>
</tr>
<tr>
<td>1</td>
<td>21024 • $T_S$</td>
<td>24576 • $T_S$</td>
</tr>
<tr>
<td>2</td>
<td>6240 • $T_S$</td>
<td>2 • 24576 • $T_S$</td>
</tr>
<tr>
<td>3</td>
<td>21024 • $T_S$</td>
<td>2 • 24576 • $T_S$</td>
</tr>
<tr>
<td>4 (for TDD only)</td>
<td>448 • $T_S$</td>
<td>4096 • $T_S$</td>
</tr>
</tbody>
</table>

Table 4.6 Random access preamble format

Preamble sequence is generated by cyclic-shifting a prime-length Zadoff-Chu (ZC) [25] [26] sequences which is also used for generation of pilot symbols. The main property for this sequence is that there is a Zero-Correlation Zone (ZCZ) between two ZC sequences derived from cyclic-shifting the same single root. The prime-length ZC sequence is defined as

$$x_{p}(n) = \exp[-j\pi n(n+1)/N_{ZC}], 0 \leq n \leq N_{ZC} - 1$$

(4.15)

Where $p$ is the ZC sequence root index and $N_{ZC}$ is the sequence length. It is specified as 839 for format 0–3 and 139 for format 4. The orthogonal preamble sequence is obtained by cyclically shifting a single root ZC sequence with the offset $N_{CS}$. Additional ZC root sequences maybe used when $N_{CS} < 64$. For each cell, a set of 64 preamble sequences is predefined and broadcast to the UEs through PBCH. There are 16 configurations of $N_{CS}$ for unrestricted and restricted
respectively. Restricted $N_{CS}$ is mainly used in High-Speed cells for avoid the impact of frequency offset by filtering some cyclic shift positions in the ZC root sequence [27].

One of the PRACH signature generation procedures is shown in Figure 4.11. For format 0–3, $N_{ZC}=839$, the size of IFFT is set to 1024, while for format 4, $N_{ZC}=139$, the size of IFFT is set to 256.

![Figure 4.11](image1.png)

**Figure 4.11** Hybrid frequency/time domain PRACH generation [27]

### 4.3.3 eNodeB PRACH Receiver

At the eNodeB side, the BS first performs CP removal and then extracts the relevant PRACH signal through a time-domain frequency shift. An $N_{FFT}$ size FFT is implemented after that following by the demapping of subcarriers. Then the BS computes PRACH Power Delay Profile (PDP) through a frequency domain periodic correlation and then performs signature detection. Figure 4.12 illustrates the basic structure of PRACH receiver, where $(\cdot)^*$ denotes the complex conjugate.

![Figure 4.12](image2.png)

**Figure 4.12** PRACH receiver structure [27]

The signature detection is consists of searching within each ZCZ defined by each cyclic shift, the PDP peaks above a detection threshold within a search window according to the cell size, collision detection, timing estimation and channel quality estimation.
For compute PRACH Power Delay Profile of the received sequence, define [27]:

\[
PDP(l) = \left| z_n(l) \right|^2 = \left| \sum_{n=0}^{N_{ZC}-1} y(n)x_n^*[(n + l)N_{ZC}] \right|^2
\]

(4.16)

Where * denotes the complex conjugate, \( y(n) \) refers to the received sequence and \( x_n(n) \) is the reference searched ZC sequence of length \( N_{ZC} \). \( z_n(l) \) denotes the discrete periodic correlation function at lag \( l \) of \( y(n) \) and \( x_n(n) \). If we define

\[
Z_u(k) = Y(k)X_u^*(k) \quad \text{for} \quad k=0,...,N_{ZC}-1
\]

(4.17)

Where \( X_u(k) = R_{X_u}(k) + jI_{X_u}(k) \), \( Y_u(k) = R_{Y_u}(k) + jI_{Y_u}(k) \), the PDP of the received sequence can be denoted as

\[
PDP(l) = \left| IDFT\{Z_u(k)\}_l \right|^2 \quad \text{for} \quad l=0,...,N_{ZC}-1
\]

(4.18)

The UE’s preamble signature will be detected if the PDP is larger than the detection threshold which is precomputed and stored in the eNodeB. Figure 4.13 illustrates the preamble signature detection based on the result of PDP computations.

![Figure 4.13 Signature Detection based on Power Delay Profile computation](image)

Collision detection can only be performed in large cell when the PDPs of two UEs are distinct from each other. For small cell, collision detection is not possible [27].

The overall complexity of the PRACH receiver can be roughly denoted by a \( N_{FFT} \) size FFT, \( N_{ZC} \) complex multiplication for computing \( Z_u(k) \), a IDFT of size \( N_{IDFT} \) which is the minimum number larger than \( N_{ZC} \) (after zero padding) and modules of complex numbers of a vector with vector size \( N_{IDFT} \).

For FDD format, \( N_{ZC}=839, N_{FFT}=N_{IDFT}=1024 \), the computational complex should be

\[
C_{\text{format}0-3}=5N_{FFT}\log_2N_{FFT}+6N_{ZC}+5N_{IDFT}\log_2N_{IDFT}+3N_{IDFT}
\]

\[
=51200+5034+51200+3072=110506 \text{ real operations.}
\]
For TDD format 4, $N_{ZC}=139$, $N_{FFT}=N_{IDFT}=256$, the complexity should be

$$C_{format0-3}=5 N_{FFT}\log_2 N_{FFT}+6 N_{ZC}+5 N_{IDFT}\log_2 N_{IDFT}+3 N_{IDFT}$$

$$=10240+834+10240+768=22082$$
real operations.

In case multiple receiving antennas are used, the complexity should be $N_r$ times the complexity listed above where $N_r$ is the number of receiving antennas.

### 4.3.4 Timing Advance Procedure

PRACH preamble signature is designed mainly for estimating a UE’s transmission timing at the eNodeB side by using the earliest detected peak of a detected signature. After the eNodeB estimated the uplink timing, it will send an 11-bit initial Timing Advance (TA) command which is included in the feedback RAR message. The TA is configured by eNodeB with a granularity of 0.52µs from 0 to 0.67ms, according to a cell radius of 100km [28]. In order to keep synchronizing with the eNodeB, TA must be updated from time to time because the propagation delay maybe changed due to the movement of the UEs. This is achieved by eNodeB monitoring any uplink signal such as Sounding Reference Signals (SRSs), Channel Quality Indicator (CQI), ACKnowledgements/Negative ACKnowledgements (ACK/NACKs) or the uplink transmission data [27]. Updating procedure is implemented in the eNodeB by initializing and issuing the TA updating command to the UEs to instruct the UEs to adjust their transmission timing. For each UE, eNodeB configures a timer which will be reset at UE every time it receives TA update command. If no TA update command is received at the UE before the timer expired, it is not allowed for any further uplink transmission for the UE and a RAC procedure is re-invoked. In LTE, typically update command will not be sent more than two times per second since fast updates are not necessary for the typical urban scenario and will cause more workload for the eNodeB.

### 4.4 LS Channel estimation for Uplink

By inserting pilot symbols which are known by both the transmitter and the receiver into the transport data frame, the receiver can estimate the whole channel by drawing all the received pilot symbols and then calculating the channel information. This information is either used for equalization for the received data or fed back to the transmitter to generate beamforming weight vector. Two estimators are mainly used for channel estimation, which are Least Square (LS) estimator and Minimum Mean Square Error (MMSE) estimator. Since LS estimator is the most straightforward estimator, we will use LS channel estimator for simplicity.

In [29], the authors give an analysis on an MIMO-OFDM system. Since only 1x1 and 1x2 SIMO configuration are used for LTE uplink [8], that is to say, only one TX antenna transmit signal from the UE to eNodeB. The eNodeB can use 1 or 2 antenna for receiving data.
Suppose there are K sub-carriers in one SC-FDMA block. Channels between TX-RX antennas couples are mutual uncorrelated. For simplicity, we assume the channel is slow fading. At a transmission time n, a binary bits stream is coded into a SC-FDMA symbol blocks. The signal on k-th subcarrier is denoted by \( X[n, k] \), where \( k = 0, ..., K - 1 \).

The received signal at RX antenna \( j \) is

\[
Y_j[n,k] = X[n,k]H_j[n,k] + N_j[n,k] \quad (4.19)
\]

where \( H_j[n,k] \) is the Channel matrix, \( N_j[n,k] \) is the additive Gaussian noise with zero mean and variance \( \sigma_n^2 \). Due to the reason that the RX antennas are independent to each other, the notation \( j \) will be omitted in the following parts.

For simplicity, we define [29]:

\[
X(n) = \text{diag}(X[n,0],X[n,1],...,X[n,K - 1]) \in \mathbb{C}^{K \times K}
\]

\[
H = [H(0) H(1) \ldots H(K-1)]^T \in \mathbb{C}^{K \times 1}
\]

\[
Y(n) = [Y(n,0) Y(n,1) \ldots Y(n,K-1)]^T \in \mathbb{C}^{K \times 1}
\]

Then Eq. (4.19) can be rewritten as

\[
Y(n) = X(n)H + N(n) \quad (4.21)
\]

For simplicity, time index \( n \) will be omitted in the following parts. Then we use a Least Square estimator to estimate the channel coefficient \( \hat{H} \). LS estimator can be denoted as

\[
\hat{H} = \arg \min_{\hat{H}} \| Y - XH \| \quad (4.22)
\]

For LTE uplink, pilot symbols are inserted into all the subcarriers. This kind of pilot is called block-type pilot. The structure of pilot symbols is shown in Figure 4.14.

If we define a \( K \times K \) matrix where the diagonal elements are \( K \) pilot symbols at time \( n \),

\[
P(n) = d \uparrow \left[ p[n,0], p[n,1], \ldots p[n,K-1] \right] \in \mathbb{C}^{K \times K} \quad (4.23)
\]

The channel matrix \( \hat{H} \) can be estimated by finding the minimum solution below:

\[
\hat{H}_{LS}[n] = \arg \min_{\hat{H}} \| Y(n) - P(n)H_{LS}[n] \| \quad (4.24)
\]

For the k-th subcarrier:

\[
\hat{H}(n,k) = \frac{Y(n,k)}{P(n,k)} \quad (4.25)
\]
By calculating $\hat{H}_{LS}(k)$ on every of the K subcarriers, channel matrix $\hat{H}_{LS}[n]$ is derived. The channel coefficient at the other SC-FDMA data symbols positions can be calculated using linear interpolation (Figure 4.15).

The linear interpolation can be denoted as

$$H[n] = \frac{H_p[n_k] - H_p[n_{k-1}]}{n_k - n_{k-1}}(n - n_{k-1}) + H[n_{k-1}]$$  \hspace{1cm} (4.26)
**Complexity Analysis**

Since the inverse of the pilot symbols can be pre calculated and stored, we will only focus on real-time computational complexity.

As discussed above, $k$-point radix-2 FFT usually need $0.5k \log_2 k$ complex multiplications and $k \log_2 k$ complex addition - all together $5k \log_2 k$ real operations. The analysis result is listed in Table 4.8.

<table>
<thead>
<tr>
<th>Complex Operation</th>
<th>Real Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complex-complex mult.</td>
<td>4 real mult. + 2 real add. = 6</td>
</tr>
<tr>
<td>Complex-real mult.</td>
<td>2 real mult.</td>
</tr>
<tr>
<td>Complex-complex add./sub.</td>
<td>2 real add./sub. = 2</td>
</tr>
</tbody>
</table>

**Table 4.7 Conversion from complex to real operations**

<table>
<thead>
<tr>
<th>Real Time Complexity</th>
<th>Function</th>
<th>Size</th>
<th>Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FFT</td>
<td>$K_{FFT}$</td>
<td>$5K_{FFT} \log_2 K_{FFT}$</td>
</tr>
<tr>
<td></td>
<td>Complex multi.</td>
<td>-</td>
<td>$K$ $6K$</td>
</tr>
</tbody>
</table>

**Table 4.8 Complexity Analysis result for LTE uplink FFT-based LS channel estimation**

($K_{FFT}$ is FFT size for a given bandwidth and usually is a power-of-two number.)

For the case LTE uplink 20MHz bandwidth, $K=1200$, $K_{FFT}=2048$, normal CP, the overall channel estimation complexity for a whole frame on pilot symbols should be

$$20\text{slot}*(6K+5K_{FFT} \log_2 K_{FFT}) = 2396800 \text{ real operations}$$

In real case, the estimation is never done so frequently on the whole frame at each pilot position, the Basestation will decide the estimation frequency depending on the mobility, for example the estimation can be performed every 5ms in case of low mobility.

For linear interpolation, the overall complexity for the whole frame with normal CP should be

$$1200*[19*(1 \text{ complex subtraction} + 1 \text{ complex-real multiplication} + 6 \text{ complex additions}) + 6 \text{ complex additions}] = 1200*316 = 379200 \text{ real operations}$$

Similarly, for extended CP the complexity should be

$$1200*[19*(1 \text{ complex subtraction} + 1 \text{ complex-real multiplication} + 5 \text{ complex additions}) + 5 \text{ complex additions}] = 1200*276 = 331200 \text{ real operations}$$
The total computational complexity including linear interpolation should be 2776000 and 2728000 real operations for normal and extended CP respectively.

4.5 Linear Equalization for LTE uplink

After estimate the channel matrix $H$, the complex-valued modulation symbol data can be equalized by using Zero Forcing or MMSE linear equalizer.

4.5.1 System Model:

As mentioned in 2.2.6, LTE uplink is a Single Input Multiple Output system with one transmit antenna and $Nr$ receive antennas, where $Nr = 1$ or 2. Let $s$ be the modulated complex-valued symbol vector to be transmitted. The modulation schemes should be QPSK (4QAM), 16QAM or 64QAM. Normalization must be performed for each symbol $s_i$ before transmitting to ensure the transmit power being independent from the modulation used. For M-QAM modulation, the complete set of transmitted symbol is denoted by $\Omega = \left\{ \pm \frac{1}{2}g, \pm \frac{3}{2}g, ..., \pm \sqrt{\frac{M-1}{2}}g, \right\}$, where $g = \sqrt{\frac{6}{M-1}}$, so the power of the transmitted complex-valued signals is normalized to 1 [35].

The basic system model is:

$$r = Hs + n$$  \hspace{1cm} (4.27)

where $r$ is the received data, $H$ is an $Nr \times 1$ the complex-valued channel vector derived from the channel estimation, $n$ is the additive noise vector at the receiver. By using linear equalizers such as Zero forcing or MMSE, the estimation of the transmitted symbol vector $s$ can be denoted by $\hat{s}$.

Zero forcing equalizer is performed at the receiver side by solving [32]:

$$\hat{s} = \arg \min_s \| Hs - r \|^2$$  \hspace{1cm} (4.28)

The solution for (4.28) is given by the pseudo inverse of $H$ [31], denoted by $H^+$

$$\hat{s} = H^+ r$$  \hspace{1cm} (4.29)

For one symbol $S(m)$ in $s$, we have:

$$\hat{S}(m) = \frac{1}{\hat{H}(m)} R(m) = \frac{H(m)}{\hat{H}(m)} S(m) + \frac{1}{\hat{H}(m)} N(m)$$  \hspace{1cm} (4.30)
Although ZF equalizer is easy to implement, it amplifies the noise \( n \) contained in the received signal vector \( s \), especially if \( H \) is not accurately estimated at the receiver. The problem is solved by a more complex linear equalizer called MMSE equalizer.

MMSE equalizer is performed at the receiver by minimize the mean square error \( E(\|\hat{s} - s\|^2) \), solving [32]:

\[
\hat{s} = \arg \min_s \|Hs - r\|^2 + \|\sigma s\|^2
\]  

(4.31)

where \( \sigma \) is the inverse SNR of the transmitted signal. The minimum-norm solution for (4.31) is given by [33], [34]:

\[
\hat{s} = Gr,
\]

\[
G = (H^H H + \sigma^2 I)^{-1} H^H
\]  

(4.32)

where \( I \) is the identity matrix. For one symbol \( S(m) \) in \( s \), we have:

\[
\hat{S}(m) = \frac{\hat{H}^*(m)}{\|\hat{H}(m)\|^2 + \hat{\sigma}^2} R(m) = \frac{\hat{H}^*(m)H(m)}{\|\hat{H}(m)\|^2 + \hat{\sigma}^2} S(m) + \frac{\hat{H}^*(m)}{\|\hat{H}(m)\|^2 + \hat{\sigma}^2} N(m)
\]  

(4.33)

The complexity for calculations for \( G \) is \( 6N_r^2N_r - 2N_tN_r \) real-valued multiply-and-accumulate (MAC) operations by using a novel algorithm [32]. For our case (\( N_t=1 \), \( N_r=1 \) or \( 2 \)) it only needs 4 MACs for uplink (1x1 SISO), 8 MACs to calculate \( G \) (1x2 SIMO). Suppose one slot of \( m \) symbols is received at each antenna, the complexity of calculating one symbol \( \hat{s}(m) \) is \( N_r \) complex MACs (without the complexity of calculating \( G \)) which equivalent to \( 4N_r \) real MACs. For one slot except pilot symbol there are 6 symbols for normal CP and 5 symbols for extended CP. The 6 or 5 symbols for one slot need \( 6N_r \) or \( 5N_r \) MACs plus \( 6N_t^2N_r - 2N_tN_r \) MACs for compute \( G \). For one Resource Block, \( N_{RB}^{SC} \) subcarriers need \( N_{RB}^{RB} \) times MACs of that for one slot. Suppose uplink has \( N_{UL}^{RB} \) Resource Blocks, for one frame transmitted data, the complexity should be

\[
N_{UL}^{RB} \cdot N_{RB}^{RB} \cdot 2 \cdot 6 \left[ (6N_r - 2N_r) + 4N_r \right] \text{MACs for normal CP}
\]

\[
N_{UL}^{RB} \cdot N_{RB}^{RB} \cdot 2 \cdot 6 \left[ (6N_r - 2N_r) + 4N_r \right] \text{MACs for extended CP}
\]

A case study is shown here: for LTE uplink 20MHz bandwidth 1x2 MIMO, 1 frame is received at each antenna. The complexity for equalized the whole frame should be:

\[
100 \cdot 12 \cdot 20 \cdot 6 \cdot (8 \times 2) = 2304000 \text{MACs for normal CP}
\]

\[
100 \cdot 12 \cdot 20 \cdot 5 \cdot (8 \times 2) = 1920000 \text{MACs for extended CP}
\]
4.5.2 Soft-Output Detection

Given the channel matrix $\hat{H}(m)$ and $\hat{\sigma}^2$, we can equalized the received signal and get complex-valued modulation symbol $\hat{S}(m)$. But the equalized data symbol may not have the right modulation value. The easiest way is use hard decision which is to round $\hat{S}(m)$ to the nearest lattice point, so that $\hat{S}(m) \in \Omega$. Another method called soft decision is to generate soft information by calculating LLR (Log-Likelihood Ratio). Since soft decision has better performance for channel coding, it is preferred instead of hard decision. Eq (4.34) defines the extrinsic LLRs of the $k$-th bit in the transmitted symbol $s$ to perform the Max A Posteriori (MAP) bit detection.

$$L(b_k) = \log \left( \frac{p(b_k = 1 \mid y)}{p(b_k = 0 \mid y)} \right)$$  \hspace{1cm} (4.34)

Since the high computational complexity of Eq. (4.34), maximum log-MAP algorithm is introduced to reduce the complexity.

$$L(b_k) = -\frac{1}{\sigma^2} \left\{ \min_{\alpha \in \Omega_k^{(0)}} \left\| \hat{s} - \alpha \right\|^2 - \min_{\alpha \in \Omega_k^{(1)}} \left\| \hat{s} - \alpha \right\|^2 \right\}$$  \hspace{1cm} (4.35)

In [36], the approximation for the $M$-QAM Gray coded [37] systems is analyzed. Consider the first digit of the constellation point, 0 corresponds to 8 points in the left half plane and 1 corresponds to another 8 points in the right half plane (Figure 4.16). The first digit of $\hat{s}_k$ can be estimated by finding the minimum distance between $\hat{s}_k$ and 8 points of the left half plane and 8 points of the right half plane respectively.
For the case $|\text{Re}\{\hat{s}_k\}| \leq 2$, which means $\hat{s}_k$ is in the square area marked with dashes, the minimum distances between $\hat{s}_k$ and the 8 points in the left half plane should be the distance between $\hat{s}_k$ and the 4 points which is the second column of the 8 points, these 4 points are denoted by $\alpha^{(2)}$. Similarly the minimum distances between $\hat{s}_k$ and the 8 points in the right half plane should be the distance between $\hat{s}_k$ and the 4 points which is the first column of the 8 points in the right half plane, these 4 points are denoted by $\alpha^{(3)}$.

\[
\begin{align*}
\min_{\alpha \in \Omega_k^{(0)}} \|\hat{s} - \alpha\|^2 &= (\text{Re}\{\hat{s}\} - \text{Re}\{\alpha^{(2)}\})^2 = (\text{Re}\{\hat{s}\} + 1)^2 \\
\min_{\alpha \in \Omega_k^{(1)}} \|\hat{s} - \alpha\|^2 &= (\text{Re}\{\hat{s}\} - \text{Re}\{\alpha^{(3)}\})^2 = (\text{Re}\{\hat{s}\} - 1)^2
\end{align*}
\]
So (4.35) can be calculated as:

\[
L(b_k) = -\frac{1}{\sigma^2} \left\{ (\text{Re}\{\hat{s}\} + 1)^2 - (\text{Re}\{\hat{s}\} - 1)^2 \right\} = -\frac{1}{\sigma^2} \left\{ 4 \text{Re}\{\hat{s}\} \right\}
\] (4.37)

Similarly, for the case \(\text{Re}\{\hat{s}\} > 2\),

\[
\min_{\alpha \in \Omega_k^{(0)}} \|\hat{s} - \alpha\|^2 = (\text{Re}\{\hat{s}\} - \text{Re}\{\alpha^{(2)}\})^2 = (\text{Re}\{\hat{s}\} + 1)^2
\]

\[
\min_{\alpha \in \Omega_k^{(1)}} \|\hat{s} - \alpha\|^2 = (\text{Re}\{\hat{s}\} - \text{Re}\{\alpha^{(4)}\})^2 = (\text{Re}\{\hat{s}\} - 3)^2
\] (4.38)

and

\[
L(b_k) = -\frac{1}{\sigma^2} \left\{ (\text{Re}\{\hat{s}\} + 1)^2 - (\text{Re}\{\hat{s}\} - 3)^2 \right\} = -\frac{1}{\sigma^2} \left\{ 8 \text{Re}\{\hat{s}\} - 8 \right\}
\] (4.39)

For the case \(\text{Re}\{\hat{s}\} < -2\),

\[
\min_{\alpha \in \Omega_k^{(0)}} \|\hat{s} - \alpha\|^2 = (\text{Re}\{\hat{s}\} - \text{Re}\{\alpha^{(1)}\})^2 = (\text{Re}\{\hat{s}\} + 3)^2
\]

\[
\min_{\alpha \in \Omega_k^{(1)}} \|\hat{s} - \alpha\|^2 = (\text{Re}\{\hat{s}\} - \text{Re}\{\alpha^{(3)}\})^2 = (\text{Re}\{\hat{s}\} - 1)^2
\] (4.40)

and

\[
L(b_k) = -\frac{1}{\sigma^2} \left\{ (\text{Re}\{\hat{s}\} + 3)^2 - (\text{Re}\{\hat{s}\} - 1)^2 \right\} = -\frac{1}{\sigma^2} \left\{ 8 - 8 \text{Re}\{\hat{s}\} \right\}
\] (4.41)

The results for \(\text{Re}\{\hat{s}\} > 2\) and \(\text{Re}\{\hat{s}\} < -2\) can be combined as:

For the case \(\|\text{Re}\{\hat{s}\}\| > 2\):

\[
L(b_k) = -\frac{1}{\sigma^2} \left\{ 8 \text{Re}\{\hat{s}\} - 8 \text{sgn}\{\text{Re}\{s\}\} \right\}
\] (4.42)

The 2\(^{nd}\), 3\(^{rd}\) and 4\(^{th}\) digit can be estimate in the similar way. The computational complexity for \(M\)-QAM gray-coded constellation \((M=4, 16, 64)\) is given in Table 4.9, 4.10, 4.11. The overall complexity for estimating one \(M\)-QAM modulated symbol is summarized in Table 4.13 in terms of equivalent additions. (In fact, the computational complexity can be further reduced by using shifting operations instead of multiplication operations.)
\[
\begin{array}{|c|c|c|c|}
\hline
k & L(b_k) & \text{Range} & \text{Complexity} \\
\hline
0 & -\frac{1}{\sigma^2} \{4 \text{Re} \{\hat{s}_k\}\} & \text{all } \text{Re} \{\hat{s}\} & 2\text{mult.} \\
1 & -\frac{1}{\sigma^2} \{4 \text{Im} \{\hat{s}_k\}\} & \text{all } \text{Im} \{\hat{s}\} & 2\text{mult.} \\
\hline
\end{array}
\]

Table 4.9 LLR Approximation for 4-QAM Gray-Coded Constellations

\[
\begin{array}{|c|c|c|c|}
\hline
k & L(b_k) & \text{Range} & \text{Complexity} \\
\hline
0 & -\frac{1}{\sigma^2} \{4 \text{Re} \{\hat{s}_k\}\} & |\text{Re} \{\hat{s}\}| \leq 2 & 2\text{mult.} \\
& -\frac{1}{\sigma^2} \{8 \text{Re} \{\hat{s}_k\} - 8 \text{sgn}(\text{Re} \{\hat{s}_k\})\} & |\text{Re} \{\hat{s}\}| > 2 & 2\text{mult.}, 1\text{sgn}, 1\text{add.} \\
1 & -\frac{1}{\sigma^2} \{8 - 4|\text{Re} \{\hat{s}_k\}|\} & \text{all } \text{Re} \{\hat{s}\} & 2\text{mult.}, 1\text{add.}, 1\text{abs.} \\
\hline
2 & -\frac{1}{\sigma^2} \{4 \text{Im} \{\hat{s}_k\}\} & |\text{Im} \{\hat{s}\}| \leq 2 & 2\text{mult.} \\
& -\frac{1}{\sigma^2} \{8 \text{Im} \{\hat{s}_k\} - 8 \text{sgn}(\text{Im} \{\hat{s}_k\})\} & |\text{Im} \{\hat{s}\}| > 2 & 2\text{mult.}, 1\text{sgn}, 1\text{add} \\
3 & -\frac{1}{\sigma^2} \{8 - 4|\text{Im} \{\hat{s}_k\}|\} & \text{all } \text{Im} \{\hat{s}\} & 2\text{mult.}, 1\text{add.}, 1\text{abs.} \\
\hline
\end{array}
\]

Table 4.10 LLR Approximation for 16-QAM Gray-Coded Constellations
<table>
<thead>
<tr>
<th>$k$</th>
<th>$L(b_k)$</th>
<th>Range</th>
<th>Complexity</th>
</tr>
</thead>
</table>
| 0   | $-\frac{1}{\sigma^2} \left\{ 4 \text{Re} \left\{ \hat{s}_k \right\} \right\}$ | $|\text{Re} \left\{ \hat{s} \right\| \leq 2$ | 2 mult.  
2 mult., 1 sgn, 1 add. 
2 mult., 1 sgn, 1 add. 
2 mult., 1 sgn, 1 add. |
|     | $-\frac{1}{\sigma^2} \left\{ 8 \text{Re} \left\{ \hat{s}_k \right\} - 8 \text{sgn} \left( \text{Re} \left\{ \hat{s}_k \right\} \right) \right\}$ | $2 < |\text{Re} \left\{ \hat{s} \right\| \leq 4$ |  
|     | $-\frac{1}{\sigma^2} \left\{ 12 \text{Re} \left\{ \hat{s}_k \right\} - 24 \text{sgn} \left( \text{Re} \left\{ \hat{s}_k \right\} \right) \right\}$ | $4 < |\text{Re} \left\{ \hat{s} \right\| \leq 6$ |  
|     | $-\frac{1}{\sigma^2} \left\{ 16 \text{Re} \left\{ \hat{s}_k \right\} - 48 \text{sgn} \left( \text{Re} \left\{ \hat{s}_k \right\} \right) \right\}$ | $|\text{Re} \left\{ \hat{s} \right\| > 6$ |  
| 1   | $-\frac{1}{\sigma^2} \left\{ 24 - 8|\text{Re} \left\{ \hat{s}_k \right\| \right\}$ | $|\text{Re} \left\{ \hat{s} \right\| \leq 2$ | 2 mult., 1 add., 1 abs.  
2 mult., 1 add., 1 abs.  
2 mult., 1 add., 1 abs.  
|     | $-\frac{1}{\sigma^2} \left\{ 6 - 4|\text{Re} \left\{ \hat{s}_k \right\| \right\}$ | $2 < |\text{Re} \left\{ \hat{s} \right\| \leq 6$ |  
|     | $-\frac{1}{\sigma^2} \left\{ 40 - 8|\text{Re} \left\{ \hat{s}_k \right\| \right\}$ | $|\text{Re} \left\{ \hat{s} \right\| > 6$ |  
| 2   | $-\frac{1}{\sigma^2} \left\{ 4|\text{Re} \left\{ \hat{s}_k \right\| - 8 \right\}$ | $|\text{Re} \left\{ \hat{s} \right\| \leq 4$ | 2 mult., 1 add., 1 abs.  
2 mult., 1 add., 1 abs.  
|     | $-\frac{1}{\sigma^2} \left\{ 24 - 4|\text{Re} \left\{ \hat{s}_k \right\| \right\}$ | $|\text{Re} \left\{ \hat{s} \right\| > 4$ |  
| 3   | $-\frac{1}{\sigma^2} \left\{ 4 \text{Im} \left\{ \hat{s}_k \right\} \right\}$ | $|\text{Im} \left\{ \hat{s} \right\| \leq 2$ | 2 mult., 1 add., 1 abs.  
2 mult., 1 add., 1 abs.  
2 mult., 1 add., 1 abs.  
|     | $-\frac{1}{\sigma^2} \left\{ 8 \text{Im} \left\{ \hat{s}_k \right\} - 8 \text{sgn} \left( \text{Im} \left\{ \hat{s}_k \right\} \right) \right\}$ | $2 < |\text{Im} \left\{ \hat{s} \right\| \leq 4$ |  
|     | $-\frac{1}{\sigma^2} \left\{ 12 \text{Im} \left\{ \hat{s}_k \right\} - 24 \text{sgn} \left( \text{Im} \left\{ \hat{s}_k \right\} \right) \right\}$ | $4 < |\text{Im} \left\{ \hat{s} \right\| \leq 6$ |  
|     | $-\frac{1}{\sigma^2} \left\{ 16 \text{Im} \left\{ \hat{s}_k \right\} - 48 \text{sgn} \left( \text{Im} \left\{ \hat{s}_k \right\} \right) \right\}$ | $|\text{Im} \left\{ \hat{s} \right\| > 6$ |  
| 4   | $-\frac{1}{\sigma^2} \left\{ 24 - 8|\text{Im} \left\{ \hat{s}_k \right\| \right\}$ | $|\text{Im} \left\{ \hat{s} \right\| \leq 2$ | 2 mult., 1 add., 1 abs.  
2 mult., 1 add., 1 abs.  
2 mult., 1 add., 1 abs.  
|     | $-\frac{1}{\sigma^2} \left\{ 6 - 4|\text{Im} \left\{ \hat{s}_k \right\| \right\}$ | $2 < |\text{Im} \left\{ \hat{s} \right\| \leq 6$ |  
|     | $-\frac{1}{\sigma^2} \left\{ 40 - 8|\text{Im} \left\{ \hat{s}_k \right\| \right\}$ | $|\text{Im} \left\{ \hat{s} \right\| > 6$ |  
| 5   | $-\frac{1}{\sigma^2} \left\{ 4|\text{Im} \left\{ \hat{s}_k \right\| - 8 \right\}$ | $|\text{Im} \left\{ \hat{s} \right\| \leq 4$ | 2 mult., 1 add., 1 abs.  
2 mult., 1 add., 1 abs.  
|     | $-\frac{1}{\sigma^2} \left\{ 24 - 4|\text{Im} \left\{ \hat{s}_k \right\| \right\}$ | $|\text{Im} \left\{ \hat{s} \right\| > 4$ |  

**Table 4.11** LLR Approximation for 64-QAM Gray-Coded Constellations

<table>
<thead>
<tr>
<th>Operations</th>
<th>Number of Equivalent Additions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addition, Subtraction</td>
<td>1</td>
</tr>
<tr>
<td>Multiplication, Division</td>
<td>1</td>
</tr>
<tr>
<td>Comparison</td>
<td>1</td>
</tr>
<tr>
<td>Absolution, Sign</td>
<td>1</td>
</tr>
</tbody>
</table>

**Table 4.12** Number of equivalent additions per operation
Table 4.13 Complexity of M-QAM per symbol in equivalent additions

<table>
<thead>
<tr>
<th>k</th>
<th>4-QAM</th>
<th>16-QAM</th>
<th>64-QAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>Total</td>
<td>4</td>
<td>18</td>
<td>34</td>
</tr>
</tbody>
</table>

For LTE uplink 20MHz bandwidth case, there are $1200_{sc} \times 20_{slot} \times 6/5_{symb} = 144000$ (normal CP) or $120000$ (extended CP) symbols in one radio frame (except pilot symbols). The computational cost for one frame is listed in Table 4.14.

<table>
<thead>
<tr>
<th></th>
<th>4-QAM</th>
<th>16-QAM</th>
<th>64-QAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal CP</td>
<td>576000</td>
<td>2592000</td>
<td>4896000</td>
</tr>
<tr>
<td>Extended CP</td>
<td>480000</td>
<td>2160000</td>
<td>4080000</td>
</tr>
</tbody>
</table>

Table 4.14 Complexity of M-QAM per frame in equivalent additions

4.6 Turbo coding

Turbo codes [38] are a class of convolutional error correction codes, whose performances are near the Shannon limit in terms of Bit Error Rate (BER). It was first introduced in 1993 by C. Bermu, A. Glavieux and P. Thitimajshima. It is built using Parallel Concatenation Convolutional codes (PCCC) [39]. The decoder works the similar way by using two serial elementary decoders. Turbo coding is an important coding method in digital communications which is the main channel coding method for both uplink and downlink of LTE.

4.6.1 Turbo encoder

A typical turbo encoder consists of two identical Recursive Systematic Convolutional (RSC) encoders with parallel concatenation separated by a random interleaver. An RSC encoder has typically 1/2 coding rate. Parallel concatenation means two RSC encoders encoding at the same time. Figure 4.17 shows a turbo encoder with coding rat 1/3. The input bits block C is first encoded by encoder 1. Since the encoder is systematic, the first output $C_0$ is equal to the input bit $C_0$. The second output is the first parity bit $P_1$ encoded by encoder 1. Encoder 2 received interleaved input bit and output the second parity bit $P_2$. The main purpose of interleave before encoder 2 is to avoid burst error and increase the minimum distance of turbo codes.
In order to drive the turbo encoder to the all-zero state, trellis termination must be implemented by inserting a number of additional zero bits after the input sequence.

### 4.6.2 LTE turbo encoder

LTE turbo encoder employs a Parallel Concatenated Convolutional Code with two 8-state constituent encoders and one turbo code internal interleaver. The coding rate is 1/3. Figure 4.18 shows the structure of the turbo encoder.

The transfer function of the 8-state constituent code for the encoder is [40]:

\[
g_0(D) = \begin{bmatrix} 1 & \frac{g_0(D)}{g_1(D)} \end{bmatrix}
\]

where

\[
g_0(D) = 1 + D^2 + D^3 \quad \text{and} \quad g_1(D) = 1 + D + D^3.
\]

The initial state of the shift registers shall be all zeros when starting to encode the input data. The output from the turbo encoder is

\[
x_k = (0), \quad z_k = (1), \quad z_k' = (2)
\]

for \( k = 0, 1, 2, ..., K - 1 \). \( K \) is the code block size from 40 to 6144 bits.

Trellis termination is performed by taking the tail bits from the shift register feedback after all information bits are encoded. Tail bits are padded after the encoding of information bits.

The first three tail bits shall be used to terminate the first constituent encoder (upper switch of Figure 4.18 in lower position) while the second constituent encoder is disabled. The last three tail bits shall be used to terminate the second constituent encoder (lower switch of Figure 4.18 in lower position) while the first constituent encoder is disabled [15]. The output bits after trellis termination should be

\[
x_K, z_K, x_{K+1}, z_{K+1}, x_{K+2}, z_{K+2}, x'_K, z'_K, x'_{K+1}, z'_{K+1}, x'_{K+2}, z'_{K+2}.
\]
4.6.3 SISO decoder

As the demodulated signals at the receiver are interfered by the channel noise, soft input is generated from the equalizer. Turbo decoder is used for error correction. Figure 4.19 shows the structure of an iterative turbo decoder. The input should be the demodulated bits with the estimated probability.

The SISO decoder first estimate the probability for each data bit. The three inputs of the SISO decoder are systematic a-priori information $\lambda a_{s1}$, systematic intrinsic information $\lambda i_{s1}$, and parity intrinsic information $\lambda i_{p1}$. The SISO decoder first calculates the extrinsic systematic information $\lambda e_{s1}$ and the soft-output information $\Lambda(d_{s1})$ for each received systematic bit. The extrinsic systematic information $\lambda e_{s1}$ from SISO decoder1 is then interleaved and fed to the SISO decoder2 as a-priori information $\lambda a_{s2}$, and also interleaved systematic intrinsic information of SISO decoder1 $\lambda i_{s1}$ as the systematic intrinsic information of SISO decoder2 $\lambda i_{s2}$. Similar to SISO decoder1, SISO decoder2 also calculate the extrinsic systematic information $\lambda e_{s2}$ and the soft-output information $\Lambda(d_{s2})$ for each systematic bit received. This process is repeated until a stopping criterion is met.
4.6.4 Turbo decoding algorithms

The most popular turbo decoding algorithms are Soft-Output Viterbi Algorithm (SOVA), which is based on APP probabilities [41], Log Maximum A Posteriori (Log-MAP) and Maximum Log-MAP (Max-Log-MAP), which are based on ML probabilities [42]. All of the algorithms are based on iterative decoding method. Log-MAP is the MAP algorithm operating in the logarithmic domain. Max-Log-MAP is a simplified version of Log-MAP algorithm.

Maximum A Posteriori estimation algorithm was first proposed by Bahl, Cocke, Jelinek and Raviv in 1974. It examines every possible path through the convolutional decoder trellis and therefore initially seems to be complex to implement for application in real systems until the discovery of turbo codes. The MAP algorithm provides not only the estimated bit sequence, but also the probabilities for each bit that has been decoded correctly.

Due to large number of multiplications, the MAP algorithm was simplified by operating the algorithm in logarithmic domain, known as Log-MAP which provides the same performance with MAP algorithm and is easy to implement. Log-MAP algorithm can be further simplified by using the approximation below. The simplified Log-MAP is called Maximum log-MAP or Max-log-MAP which using the approximation shown in Eq. (4.45). The complexity of Max-Log-MAP is less than log-MAP but the performance is slight down due to the approximation involved.

\[ \ln(e^x + e^y) \approx \max(x, y) \]  

(4.45)
Differing from the classical Viterbi algorithm, Soft output Viterbi algorithm uses a modified path metric which takes into account the a priori probabilities of the input symbols, and produces a soft output indicating the reliability of the decision [43].

### 4.6.5 Complexity of turbo decoding algorithms

In [44], a complexity analysis of three turbo decoding algorithms log-MAP, max-log-MAP and SOVA was presented with the assumption that logical and mathematical operations have similar complexity. A thorough analysis was given in [45], where each logical and mathematical operation is quantified as a number of equivalent additions.

<table>
<thead>
<tr>
<th>Operations</th>
<th>Log-MAP</th>
<th>Max-Log-MAP</th>
<th>SOVA</th>
</tr>
</thead>
<tbody>
<tr>
<td>max ops</td>
<td>$5 \cdot 2^M - 2$</td>
<td>$5 \cdot 2^M - 2$</td>
<td>$3(M+1) + 2^M$</td>
</tr>
<tr>
<td>additions</td>
<td>$15 \cdot 2^M + 9$</td>
<td>$10 \cdot 2^M + 11$</td>
<td>$2 \cdot 2^M + 8$</td>
</tr>
<tr>
<td>mult. by ± 1</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>bit comps</td>
<td></td>
<td></td>
<td>6(M+1)</td>
</tr>
<tr>
<td>look-ups</td>
<td>$5 \cdot 2^M - 2$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall Complexity</td>
<td>$25 \cdot 2^M + 13$</td>
<td>$15 \cdot 2^M + 17$</td>
<td>$3 \cdot 2^M + 9M + 25$</td>
</tr>
</tbody>
</table>

**Table 4.15** Complexity for turbo decoding with rate 1/2 [44]

Table 4.15 shows the complexity of three algorithms, where $M$ is the memory order. From complexity point of view, Log-MAP has the nearly two times complexity as SOVA does. The complexity of Max-Log-MAP is less than Log-MAP due to the approximation, but still higher than SOVA. From implementation point of view, MAP algorithms are more suited to parallel processing [46].
Chapter 5

Hardware discussion

Baseband Architecture

Wireless communication networks have operated in many forms for more than 50 years. It keeps evolving since the introduction of cellular mobile telephone networks in Australia in 1987. The design goal for mobile base station is to use reconfigurable SDR based hardware to support multiple wireless communication standards and try to reduce the cost in updating the BS.

5.1 SDR Base Station Architecture

There are four main modules in a typical SDR base station (Figure 5.1): radio frequency (RF), baseband, control and transmission. The RF module receives signals then converts them into digital data (ADC) and converts signals from digital data then transmits (DAC). The baseband module processes the encoded signal before transmitting/receiving it from/to the core network through the transmission module. Control module is to coordinate other three modules.

Figure 5.1 Base Station Architecture from [12]
(RP - Reference points, BB - Baseband)
The baseband module should contain the following processing module:

- CRC
- Code Block Segmentation / De-segmentation
- Channel coding (Turbo coding / decoding)
- Rate matching / De-matching
- Code Block Concatenation / De-concatenation
- Channel Interleaver / De-interleaver
- Scrambler / Descrambler
- Transform Precoding / Decoding
- Mapping / Demapping to / from physical resources
- Demodulation Reference signal generation
- Modulation (QPSK, 16QAM, 64QAM) / Soft Demapping
- SC-FDMA Baseband Signal Generation
- Channel estimation & Equalization
- Timing & Frequency Synchronization
- MIMO / diversity

**Figure 5.2** eNodeB Baseband High Level Architecture of ARICENT solutions [47]
Figure 5.2 briefly illustrates the LTE baseband architecture [47]. According to ARICENT solutions, the baseband / LTE PHY library is independent of both hardware and OS (Operating System). This soft-defined architecture will greatly facilitate OEMs to customize, optimize, port and integrate the baseband physical library to any platform and also helps to reduce the cost and shorten the time to market.

5.2 Evolution of Base Station Architectures

The base station architecture is evolving from a modulation-specific to software-defined architecture. In a conventional base station, the baseband and RF parts of the transceiver are usually physically close to each other. However, a significant amount RF power is wasted in cables; because it is not being generated close enough to the antenna. Two alternative topologies are used to solve the problem: distributed base station architecture and orphaned RF networks (also known as ‘BS hoteling’) [12].

Distributed base station architecture splits the RF transceivers from the rest of the base station and relocates them next to their associated antennas to achieve minimal transmission power loss.

Co-location of several base stations in a central location (the hub) is known as base station hoteling. BS hoteling offers cost savings because it reduces footprint and eliminates climate control at the cell tower.

To achieve higher data rates radio air-interface standards are continuously evolving through the introduction of advanced baseband processing techniques such as adaptive modulation and coding, space-time coding, beamforming, and MIMO antenna techniques. The main advantages of adaptive antenna system are interference cancellation and increased system capacity. However it has some drawbacks due to the complex processing algorithms, multiple RF power amplifiers, feeder cables and calibration systems which will definitely bring extra cost for the operator. The SDR technologies make it possible to realize the smart antenna system because it mitigates some of the cost, size, cabling, and calibration issues [12].

Next generation networks topology for LTE is flat network architecture. It combines several functions of a base station into a single node. As the SDR base station evolving, higher capacity will not be the only requirement, flexible services and the multiple communication standards will also be provided.

5.3 Hardware discussion

In order to achieve higher data transfer rate, complex signal processing techniques such as MIMO and radio techniques such as OFDM and SC-FDMA are employed in 3GPP LTE specifications. It requires a sound grasp of every hardware signal processing technologies to
meet the computation requirements. As the standards progress, wireless base station OEMs have to choose a powerful, cost-effective way to meet the computational capacity these systems demand. Software defined radio (SDR) platform is the best scheme for most network operator. These kinds of systems usually employ a mix of FPGAs, ASICs and DSPs and are compatible to multi-standard such as GSM, Wimax and LTE. As mentioned before, we will focus on the eNodeB hardware design of physical layer for LTE uplink below.

To partition the physical layer processing of LTE system and pick the right foundation architecture, the differences among ASICs, FPGAs and DSPs must be clear. Table 5.1 lists the main differences among them.

<table>
<thead>
<tr>
<th></th>
<th>ASIC</th>
<th>FPGA</th>
<th>DSP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Characteristic</strong></td>
<td>For the needs of specific users. Compared with general purpose IC, ASIC has smaller size, lower power consumption, improved reliability and performance, enhanced security and lower cost.</td>
<td>FPGA significantly shorten the design time of digital circuits, reduce PCB space and improve system reliability.</td>
<td>Specialized for digital signal processing. DSP has high processing speed, higher system integration density, and software programmability with flexibility.</td>
</tr>
<tr>
<td><strong>Distinction</strong></td>
<td>The boundaries between ASIC and FPGA are becoming blurred.</td>
<td>Compared to ASIC, the use of FPGA is flexible. Users can change its logic function whenever they want.</td>
<td>DSP is mainly used for computing, such as encryption/decryption, modulation/demodulation. The advantage of DSP is the powerful data-processing capacity and higher speed. Using VHDL programming, FPGA is flexible and can easily debug, reprogram.</td>
</tr>
<tr>
<td><strong>Application Area</strong></td>
<td>For specific users and specific electronic system.</td>
<td>FPGA can achieve any functions of any digital devices.</td>
<td>DSP applications are mainly for 3C (Communication, Computer, Consumer) areas.</td>
</tr>
</tbody>
</table>

**Table 5.1** Differences among ASIC, FPGA and DSP

**ASIC**: an Application-Specific Integrated Circuit is an integrated circuit customized for specific use. Usually it is not for general-purpose use [48]. For specific applications, ASICs give the best performance but they are less of flexibility. It cannot handle any changes of the design parameters. High NRE (non-reoccurring engineering) costs and inflexibility make ASICs not a good choice than ideal.

**FPGA**: a Field-Programmable Gate Array is a semiconductor device that can be configured by the customer or designer after manufacturing [49]. FPGAs provide similar performance to ASICs with flexibility. However, both FPGAs and ASICs are complex to design. Any changes of the design parameters may result the reprogramming design. The price for FPGA is very high which makes them unsuitable for high volume applications. FPGAs are usually used as the complement of DSPs.

**DSP**: a Digital Signal Processor is a specialized microprocessor designed specifically for digital signal processing which is generally used in real-time computing [50]. Compared to general
purpose processors, DSPs are optimized for signal processing applications, especially for video and audio coding/decoding.

A DSP processor usually quickly performs many mathematical operations on a set of data. But it is designed for some certain operation task and can only be customized to some extent. Unlike DSPs, FPGAs are more flexible because they are programmable. This feature of FPGA makes it good for software defined radio platform. Combing FPGAs and DSPs makes the whole system flexible, bug free and even compatible to different network standards.

For the wireless SDR system, partitioning strategy depends on system configuration, bandwidth, processing requirements and the number of transmit and receive antennas. At the same time, programmability, integration, development cycle, performance and power consumption are five important criteria on the choice of devices.

The eNodeB baseband physical layer processing flow for LTE uplink can be roughly categorized into bit-level processing and symbol-level processing. According to the computational complexity, the baseband processing chain of the eNodeB for LTE uplink is shown in Figure 5.3. The computational intensive part is marked with yellow block.

![Figure 5.3 DSP/FPGA partitioning for LTE uplink SC-FDMA systems (eNodeB Perspective)](image)
5.3.1 Bit-Level Processing

On the transmitter side, bit-level processing includes randomization, turbo coding, interleaving, constellation mapping. On receiver side, bit-level processing is a little different in FEC part which includes deinterleaving, Turbo decoding, convolutional decoding, rate-matching and CRC check. Due to the characteristic of bit-level processing and its heavy computational complexity, this part is either implemented using FPGA or ASIC rather than DSP which will cause overflow of system bandwidth, although it can also manage to decode. For early release, FPGA provides enough performance while consuming higher power and higher price-per-unit. After several iterations, later releases can be implemented using ASIC for the sake of power efficiency and lower price-per-unit. Since H-ARQ is introduced in LTE, large soft-buffer memory is required for the up-link. Except turbo decoding, all bit-level processing are not computationally intensive which can be handled by DSPs for their low computational complexity.

5.3.2 Symbol-Level Processing

Sub-channelization / de-sub-channelization, channel estimation / equalization and cyclic prefix insertion / removal processing, time-to-frequency / frequency-to-time domain conversion are the main symbol-level processing.

DFE: Digital front-end (DFE) handles the filtering and sample rate conversion of the baseband signals after the analog-to-digital converter (ADC). Due to the high rate of data processing and relatively fixed requirement specification, DFE is usually implemented using ASIC for the sake of silicon and power efficiency.

FFT and IFFT functions are computational intensive which involving complex multiplications at high speeds, they can be implemented either on DSPs or FPGAs.

Synchronization: Different UE need to be synchronized to the eNodeB. Synchronization is achieved by exploiting the synchronization signals and reference signals. The load of synchronization is less intensive compared to other part which can be mapped to DSP hardware. After all, efficient address calculation and memory subsystem is needed for the extraction of synchronization and reference signals.

Channel estimation and equalization are usually vendor specific which are also computational intensive and involve more control-oriented algorithms that are better performed on FPGAs.

For channel estimation, as shown in Table 2.2, for LTE uplink with 20 MHz bandwidth, 100RBs is used, $K=1200$, $K_{FFT}=2048$, the real time computational complexity for one radio frame is shown in section 4.4:

$$C_{estimation} = \frac{2776000}{2728000} \text{ real operations (normal CP / extended CP)}$$
If two receiving antennas are used, the complexity of channel estimation for eNodeB should be 2 times.

For equalization, as discussed in section 4.5, the computational complexity of one radio frame equalization for LTE uplink with 20 MHz bandwidth should be:

For normal CP:
\[ C_{\text{equalization}} = 2304000 \text{ MACs} = 4608000 \text{ real operations} \]

For extended CP:
\[ C_{\text{equalization}} = 1920000 \text{ MACs} = 3840000 \text{ real operations} \]

Because matrix multiplications and decompositions are involved in it, MIMO techniques are also computationally intensive. It’s better to leave them to FPGAs.

Demodulation: Depending on whether MIMO is supported, the demodulation of the received data subcarriers requires different amount computing power. This part is also vendor dependent which can be mapped to either DSP or ASIC.

For soft demapping, according to Table 4.14, the computational complexity of 64-QAM soft demapping for LTE uplink 20 MHz bandwidth should be:

For normal CP:
\[ C_{\text{soft demapping}} = 4896000 \text{ real operations} \]

For extended CP:
\[ C_{\text{soft demapping}} = 4080000 \text{ real operations} \]
Chapter 6
Conclusions and Future work

6.1 Conclusions

In this thesis, we studied the physical layer of LTE including several techniques such as OFDM, SC-FDMA and MIMO which are new to the cellular system. And then we discussed about two different duplex schemes of 3GPP LTE. Basically, maximizing the commonality of TDD and FDD transmission modes is one of LTE design principles [27]. The differences between them are mainly in the physical layer rooted from the different frame structures, while there is no big difference in higher layer such as MAC and RLC layer.

To achieve high data rate, the base stations for LTE system should have more real-time computational capability. The complexity analysis in this thesis is mainly focus on channel estimation, demodulation and turbo decoding of LTE uplink physical layer. At bit-level processing, real time turbo decoding is the most computational intensive part for the whole system. At symbol-level processing, channel estimation, demodulation including equalization and soft demapping and FFT/IFFT modules are computational intensive. Besides, the synchronization for LTE uplink is briefly introduced in this thesis. Since the UEs need to keep synchronizing with the BS from time to time, it will also bring significant workload to the eNodeB which cannot be ignored. At last, we also briefly discuss the system consideration for hardware design of the base station.

6.2 Future Work

This thesis is only the first step of understanding the LTE physical layer. According to the standard, many aspects are not considered. We use a LS estimator for channel estimation without taking the Doppler spread into consideration. More estimators for example MMSE estimator for high mobility conditions must be studied in the further work. Since the complexity for turbo decoding is very high, more analysis needs to be done in order to find the optimized algorithms to reduce the complexity. For LTE uplink SC-FDMA multiuser multiplexing scheme, there is an algorithm to calculate the several variable-length FFT at once by a fixed length FFT [51], of which the implementation aspects can be further investigated.
Moreover, since we only focus on physical shared channels for uplink which are mainly used for data transmission from UE to BS, other channels such as control channels, broadcast channel, multicast channel, control format indicator channel and Hybrid ARQ indicator channel should be studied in the future work.
Bibliography


[18] Ilya Pollak, “Digital Signal Processing with Applications”, Fall 2004, pp78, Figure 1.39


[40] Y. Huang, “VLSI implementation of key components in a mobile broadband receiver”, May 2009


Upphovsrätt

Detta dokument hålls tillgängligt på Internet — eller dess framtida ersättare — under en längre tid från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

Copyright

The publishers will keep this document online on the Internet — or its possible replacement — for a considerable time from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for his/her own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: http://www.ep.liu.se/

© Ningning Guo