

# Uplink Baseband Encoder Implementation

Carl Berglund

Luleå University of Technology  
MSc Programmes in Engineering  
Electrical Engineering  
Department of Computer Science and Electrical Engineering  
Division of Signal Processing

---

# **Uplink Baseband Encoder Implementation**

**Carl Berglund**

Ericsson - Lindholmen, Gothenburg  
Baseband Processing Department

June - November, 2008

---



---

# ABSTRACT

---

In the new Information Technology (IT) society demands on having fast transmission speeds are growing. Previous generations of wireless mobile networks are only made for phone calls and slow data rate services. With the 3<sup>rd</sup> generation (3G) technologies the transmission speeds are pushed up to follow today's needs. One of these technologies is Universal Mobile Telecommunications System (UMTS) which had its first specification released in 2000 by the 3<sup>rd</sup> Generation Partnership Project (3GPP) collaboration group.

To verify the baseband processing of an UMTS base station a test environment is used. Today the test environment is generating sampled antenna data in an offline fashion. The data will include information that is not used all the time and this creates unnecessary interference in the system. Due to limited amount of memory the data has to be repeated and this gives a poor statistical confidence. To generate long and unique test cases and decrease the interference a real-time system is required.

The purpose of this master's thesis is to systemize a real-time encoder for the uplink and implement the baseband encoding for this encoder. The implementation is to be done on existing hardware. It should handle signaling and conversational speech services provided by the UMTS standard and it will be investigated how to divide the functionality between a Digital Signal Processor (DSP) and an Field-Programmable Gate Array (FPGA).

A big part of this master's thesis has been to study the 3GPP standard uplink physical layer, Wideband Code Division Multiple Access (WCDMA) and to gain understanding of how the UMTS mobile network is built up. Before implementation on hardware, simulations in MATLAB were done to gain knowledge of the coding algorithms and to see that the functionality of the system worked correctly.

The baseband encoder was implemented so that all encoding steps up to scrambling are done by a DSP and the scrambling is done by an FPGA. The encoder has support for Dedicated Channel (DCH) users and can encode radio frames within given time margins.

Keywords: 3G, UMTS, WCDMA, DCH, DSP, FPGA, Real-Time Encoder



---

# PREFACE

---

This master's thesis is the final part of the Master of Science programme in Electrical Engineering at Luleå University of Technology. It has been carried out at Ericsson, Lindholmen in Gothenburg from June to November, 2008. The assignment was to implement a real-time baseband encoder for the uplink on existing hardware.

This project has given me a great experience and in the best way prepared me for the future working life that is to come.

During the project I have been in contact with many interesting people that have supported me along the way. First of all I would like to thank my supervisors, Stefan Davidsson at Ericsson and Johan Carlson at Luleå University of Technology. I would also like to thank everyone at Ericsson who has supported and helped me.

I especially would like to mention Matts-Ola Wessman for support in signal processing and uplink related questions, Ulf Pettersson who has helped me with the Digital Signal Processor (DSP), Henrik Hågebrandt who has helped me with VHDL and Ingvar Andersson for giving me the opportunity to do this master's thesis.

*Gothenburg, November 2008*



*Carl Berglund*



---

# ABBREVIATIONS

---

1G	1 <sup>st</sup> generation
2G	2 <sup>nd</sup> generation
3G	3 <sup>rd</sup> generation
3GPP	3 <sup>rd</sup> Generation Partnership Project
ALCAP	Access Link Control Application Part
A/D	Analog-to-Digital
BCH	Broadcast Channel
BER	Bit Error Ratio
BFN	Node B Frame Number
BP	Board Processor
BPSK	Binary Phase Shift Keying
CCPCH	Common Control Physical Channel
CCS	Code Composer Studio
CCTrCH	Coded Composite Transport Channel
CDMA	Code Division Multiple Access
CFN	Connection Frame Number
CN	Core Network
CPICH	Common Pilot Channel
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CS	Circuit Switched
D/A	Digital-to-Analog
DBP	Downlink Baseband Processing
DCH	Dedicated Channel
DL	Downlink
DPCCH	Dedicated Physical Control Channel
DPCH	Dedicated Physical Channel
DPDCH	Dedicated Physical Data Channel

---

DSP	Digital Signal Processor
DSCDMA	Direct-Sequence Code Division Multiple Access
EMIF	External Memory Interface
FACH	Forward Access Channel
FBI	Feedback Information
FDD	Frequency Division Duplex
FDMA	Frequency Division Multiple Access
FEC	Forward Error Correction
FIFO	First In, First Out
FP	Frame Protocol
FPGA	Field-Programmable Gate Array
GSM	Global System for Mobile communications
HDL	Hardware Description Language
HPSK	Hybrid Phase Shift Keying
HSPA	High Speed Packet Access
IMT-2000	International Mobile Telecommunications 2000
IP	Internet Protocol
ISI	InterSymbol Interference
IT	Information Technology
ITU	International Telecommunication Union
LNA	Low Noise Amplifier
MAC	Medium Access Control
MRC	Maximal-Ratio Combining
MAI	Multiple Access Interference
NMT	Nordic Mobile Telephone
OS	Operating System
OSE	Operating System Embedded
OSEck	OSE Compact Kernel
OVSF	Orthogonal Variable Spreading Factor
PCCC	Parallel Concatenated Convolutional Code
PCH	Paging Channel
PDU	Product Development Unit
PhCH	Physical Channel
PS	Packet Switched
QDMA	Quick Direct Memory Access
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RAB	Radio Access Bearer
RACH	Random Access Channel
RAN	Radio Access Network
RBS	Radio Base Station

---

RF	Radio Frequency
RLC	Radio Link Control
RNC	Radio Network Controller
RNS	Radio Network Sub-system
RRC	Radio Resource Control
RSC	Recursive Systematic Convolutional
RTOS	Real-Time Operating System
SCH	Synchronization Channel
SDRAM	Synchronous Dynamic Random Access Memory
SFN	System Frame Number
SIR	Signal-to-Interference Ratio
SF	Spreading Factor
SMS	Short Message Service
SNR	Signal-to-Noise Ratio
SRB	Signaling Radio Bearer
RRC	Root-Raised Cosine
TB	Transport Block
TBS	Transport Block Set
TDD	Time Division Duplex
TDMA	Time Division Multiple Access
TF	Transport Format
TFI	Transport Format Indicator
TFCI	Transport Format Combination Indicator
TPC	Transmit Power-Control
TrCH	Transport Channel
TTI	Time Transmission Interval
TXAD3	TX Adapter Card 3
UBP	Uplink Baseband Processing
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
UTRAN	UMTS Terrestrial Radio Access Network
VHDL	VHSIC Hardware Description Language
VHSIC	Very-High-Speed Integrated Circuits
WCDMA	Wideband Code Division Multiple Access



---

# CONTENTS

---

CHAPTER 1: INTRODUCTION	1
1.1 Background . . . . .	1
1.2 Purpose . . . . .	1
1.3 Confines . . . . .	2
CHAPTER 2: THEORY	3
2.1 Cellular networks . . . . .	3
2.2 Mobile phone network history . . . . .	4
2.3 3G . . . . .	4
2.3.1 3GPP . . . . .	4
2.3.2 UMTS . . . . .	5
2.3.2.1 Radio Access Bearer . . . . .	6
2.3.3 Node B . . . . .	6
2.3.4 UTRAN . . . . .	6
2.4 WCDMA . . . . .	8
2.4.1 General . . . . .	8
2.4.2 Spreading . . . . .	10
2.4.2.1 Synchronous . . . . .	11
2.4.2.2 Asynchronous . . . . .	12
2.4.3 Multipath and rake receivers . . . . .	13
2.4.4 Near/far problem and power control . . . . .	15
2.4.5 Handovers . . . . .	16
2.5 Physical layer . . . . .	17
2.5.1 General . . . . .	17
2.5.2 Medium Access Control . . . . .	18
2.5.3 Radio Resource Control . . . . .	18
2.5.4 Synchronization . . . . .	19
2.5.5 Transport blocks . . . . .	19
2.5.6 Radio frames . . . . .	20
2.5.7 Channels . . . . .	20
2.5.7.1 DCH . . . . .	22
2.5.7.2 DPDCH . . . . .	22
2.5.7.3 DPCCH . . . . .	22
2.5.8 Channel coding and multiplexing for uplink . . . . .	23

---

2.5.8.1	CRC attachment . . . . .	24
2.5.8.2	Transport block concatenation / code block segmentation	24
2.5.8.3	Channel coding . . . . .	24
2.5.8.4	Radio frame equalization . . . . .	25
2.5.8.5	1 <sup>st</sup> interleaving . . . . .	25
2.5.8.6	Radio frame segmentation . . . . .	26
2.5.8.7	Rate matching . . . . .	26
2.5.8.8	TrCH multiplexing . . . . .	26
2.5.8.9	Physical channel segmentations . . . . .	26
2.5.8.10	2 <sup>nd</sup> interleaving . . . . .	26
2.5.8.11	Physical channel mapping . . . . .	27
2.5.9	Modulation . . . . .	27
2.5.9.1	Modulation mapping . . . . .	27
2.5.9.2	HPSK . . . . .	27
2.5.10	Spreading . . . . .	28
2.5.10.1	Channelization . . . . .	28
2.5.10.2	Scrambling . . . . .	28
2.5.11	Channel multiplexing and I/Q mapping . . . . .	29
2.5.12	Mixing . . . . .	30
CHAPTER 3: TEST ENVIRONMENT		31
3.1	TXAD3 . . . . .	31
3.1.1	DSP . . . . .	31
3.1.2	FPGA . . . . .	31
3.1.3	Operating system . . . . .	32
CHAPTER 4: METHOD		33
4.1	Systemization . . . . .	33
4.1.1	Baseband encoding . . . . .	33
4.1.2	TX analog . . . . .	33
4.1.3	RX analog . . . . .	34
4.2	Implementation of the baseband encoder . . . . .	34
4.2.1	MATLAB simulations . . . . .	34
4.2.2	Division of functionality . . . . .	34
4.2.3	Software . . . . .	35
4.2.3.1	Code structure . . . . .	35
4.2.3.2	Bit vectors and matrices . . . . .	35
4.2.3.3	State variables . . . . .	35

---

4.2.3.4	Users . . . . .	36
4.2.3.5	Encoding procedure . . . . .	36
4.3	Channel coding and multiplexing . . . . .	38
4.3.1	DCH . . . . .	38
4.3.1.1	Randomly generated data . . . . .	38
4.3.1.2	CRC attachment . . . . .	38
4.3.1.3	Code block segmentation . . . . .	38
4.3.1.4	Channel coding . . . . .	38
4.3.1.5	Radio frame equalization . . . . .	39
4.3.1.6	1 <sup>st</sup> interleaving . . . . .	39
4.3.1.7	Radio frame segmentation . . . . .	39
4.3.1.8	Rate matching . . . . .	39
4.3.2	TrCH Multiplexing . . . . .	39
4.3.3	DPDCH . . . . .	39
4.3.3.1	2 <sup>nd</sup> interleaving . . . . .	39
4.3.3.2	Physical channel mapping . . . . .	39
4.3.4	DPCCH . . . . .	39
4.3.5	DPCH . . . . .	40
4.3.5.1	Modulation mapping . . . . .	40
4.3.5.2	Channelization . . . . .	40
4.3.5.3	Channel multiplexing and I/Q mapping . . . . .	40
4.3.5.4	Scrambling . . . . .	40
4.4	Confines . . . . .	40
4.5	Verification . . . . .	41
CHAPTER 5: RESULT		43
5.1	Baseband encoder . . . . .	43
5.2	Analysis . . . . .	43
5.3	Conclusion . . . . .	44
CHAPTER 6: DISCUSSION		45
6.1	Design choices . . . . .	45
6.2	Future work . . . . .	46
APPENDIX A:ERROR CODING		47
1.0.1	CRC . . . . .	47
1.0.2	FEC . . . . .	48
1.0.2.1	Convolutional codes . . . . .	48

---

1.0.2.2	Turbo codes . . . . .	49
1.0.3	Interleaving . . . . .	50
APPENDIX B:PROTOCOLS		51
APPENDIX C:CHARTS		53

---

# CHAPTER 1

---

## Introduction

### 1.1 Background

The “Baseband Processing” department within Product Development Unit (PDU) “Wideband Code Division Multiple Access (WCDMA) Radio Access Network (RAN)”, is responsible for the design, development and maintenance of the Downlink Baseband Processing (DBP) and Uplink Baseband Processing (UBP) subsystem for Ericsson’s 3<sup>rd</sup> generation (3G) base stations. They are active in systemization, software design, platform design and Integration & Verification.

To verify the baseband processing of an 3G base station a test environment is used. Today the test environment is generating sampled antenna data in an offline fashion. Everything is prepared in advance, therefore the data will include information that is not used all the time and this creates unnecessary interference in the system. Another disadvantage is that the length of the data is quite short due to limited amount of memory, therefore it has to be repeated which gives a poor statistical confidence in the test runs. To generate long and unique test cases and decrease the interference a real-time system is required.

### 1.2 Purpose

The purpose of this master’s thesis is to systemize a real-time encoder for the uplink and implement the baseband encoding for this encoder. The application is to be implemented on an adapter card which consists of Digital Signal Processors (DSPs) and Field-Programmable Gate Arrays (FPGAs), divide functionality between these hardware as well as specifying the applications interface towards the surrounding test environment. The work will include studying the 3<sup>rd</sup> Generation Partnership Project (3GPP) standard

uplink physical layer and designing the application after these specifications.

### **1.3 Confines**

Due to the complex structure of the air interface the solution will be limited to handle only signaling and conversational speech services provided by the Universal Mobile Telecommunications System (UMTS) standard and restrict the functionality of the application to a practicable level. It shall still be designed so that it can be extended to handle the tests of other services as well, such as interactive Packet Switched (PS) and streaming data services.

---

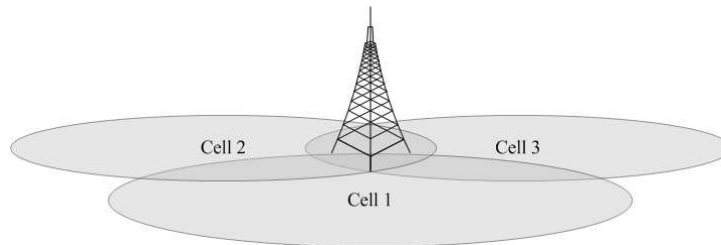
# CHAPTER 2

---

## Theory

### 2.1 Cellular networks

Cellular networks[1] consist of a number of radio cells, that each is served by a cell site (base station), Figure 2.1. The cells are distributed to give a wider coverage area than just one single cell. The mobile phone network is a cellular network.



*Figure 2.1: Cell site with three cells*

To make a cellular network operate, there has to be a standardized method to distinguish between signals from different transmitters. At the moment there exists two such methods, Frequency Division Multiple Access (FDMA) and Code Division Multiple Access (CDMA).

Two important features of a cellular network are: a broadcast signal, that is used to send information to multiple receivers and handovers, that enables switching between different cells.

The increased capacity in cellular networks comes from the fact that they can reuse

the access medium between different cell sites. This includes the frequency for FDMA and the codes for CDMA.

## 2.2 Mobile phone network history

The 1<sup>st</sup> generation (1G) mobile phone standard was introduced in the 1980's. This technology used analog signals, it was Circuit Switched (CS) and it only carried voice traffic. One example of such a standard is Nordic Mobile Telephone (NMT).

The 2<sup>nd</sup> generation (2G) was commercially launched as the Global System for Mobile communications (GSM) standard in 1991. This technology is completely digital. It uses Time Division Multiple Access (TDMA) and CDMA as air interfaces and are both CS and PS. The main benefits with 2G are that the channels can be encrypted, it is more efficient on the frequency spectrum and it can use data services like SMS.

The 3G is a standard under International Mobile Telecommunications 2000 (IMT-2000)<sup>1</sup> created by International Telecommunication Union (ITU) in 1999. It was commercially launched by NTT DoCoMo<sup>2</sup> in Japan 2001. The 3G technology used throughout this report is UMTS. It uses WCDMA as air interface, one of six air interfaces available. The third generation networks can use both CS and PS services. 2.5G is a standard that also exists, but the line between 2G and 2.5G is quite hazy.

## 2.3 3G

### 2.3.1 3GPP

3GPP is a collaboration agreement that was established in 1998. It brings together a number of telecommunication standards bodies from different countries which are known as organizational partners. These are ETSI (Europe), ARBI (Japan), TTA (Korea), T1P1/TR46.1 (USA) and CWTS (China). All partners have agreed upon the IMT-2000 RAN based on WCDMA, IMT-2000 core network based on evolved GSM and 3GPP UMTS Terrestrial Radio Access Network (UTRAN) as the IMT-2000 RAN proposal with the largest support.

The main goal of 3GPP is to produce technical specifications and technical reports for a 3G mobile systems using WCDMA and enhanced GSM core networks. It is also responsible for the future specification work of GSM.

3GPP standards are referred to as releases. Release '99 came in 2000 and has the widest dispersion. Here the first UMTS networks incorporating WCDMA as air interface are specified.

---

<sup>1</sup>The specification for all 3G systems available in the world.

<sup>2</sup>The largest telecommunications operator in Japan.

## 2.3.2 UMTS

UMTS is one of the 3G cell phone technologies. It uses WCDMA as air interface and is standardized by the 3GPP.

An UMTS network consist of three interacting domains. Core Network (CN), UTRAN and User Equipment (UE), Figure 2.2. The core network provides switching, routing and transit for user traffic. UTRAN connects UEs to the CN and UEs are the mobile nodes.

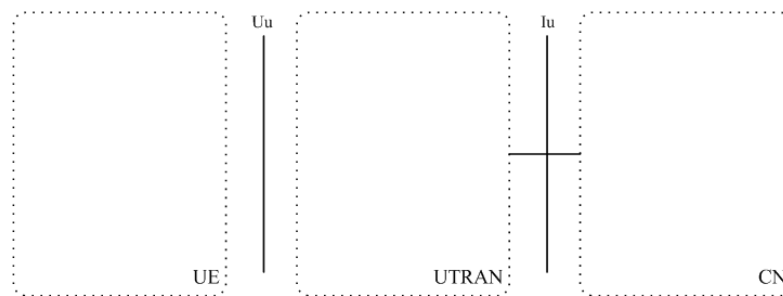


Figure 2.2: UMTS

UMTS uses two 5 MHz wide physical channels. One for the uplink and one for the downlink. The carrier wave frequency for these channels differs between different countries and technologies, Figure 2.3.

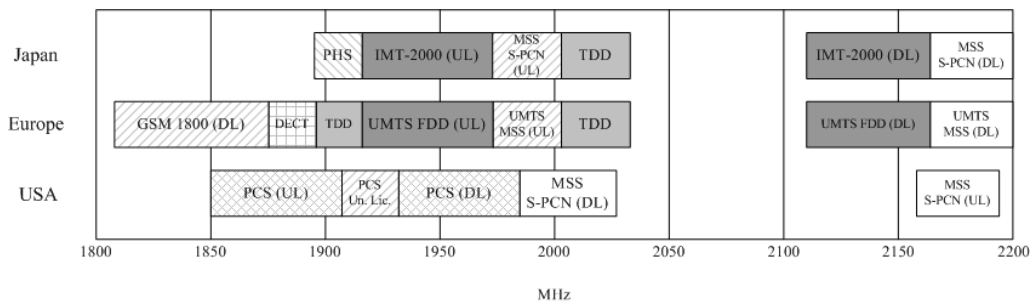


Figure 2.3: IMT-2000 spectrum allocations

Release '99 of 3GPP theoretically supports transfer rates up to 14 Mbit/s, but in the network, a rate of 384 kbit/s on CS and 2 Mbit/s on PS connections can be expected.

Newer releases have features such as High Speed Packet Access (HSPA). It will give transfer rates up to 7.2 Mbit/s on the downlink and 5.76 Mbit/s on the uplink.

### 2.3.2.1 Radio Access Bearer

A Radio Access Bearer (RAB) is the connection segment between the UE and the CN to support Quality of Service (QoS) for UMTS bearer services. Each RAB is mapped onto one or more Radio Bearers and each Radio Bearer is mapped onto one Radio Link Control (RLC).

The RAB connection can have different characteristics in order to match the requirements for different UMTS bearers. This means that an application requests the bearer depending on its demands.

The RABs are set up with an establishment, taken down with a release and can be changed with a reconfiguration.

### 2.3.3 Node B

A Node B is the base station in UMTS technology. They are stationary transceivers that are the link between UEs and the Radio Network Sub-system (RNS). A Node B has the following features:

- Air interface Transmission / Reception
- WCDMA physical channel coding
- Modulation / Demodulation
- Micro diversity
- Error handling
- Closed loop power control

They can serve several cells depending on the configuration and type of antenna. They are constructed to have minimum functionality and let the Radio Network Controller (RNC) does most of the work, but with later releases of 3GPP that includes HSPA more work is done by the Node B to lower response times.

Physically the Node B contains a cabinet, an antenna mast and an actual antenna. The cabinet contains power amplifiers, digital signal processors and back-up batteries. Hereafter an UMTS base station will always be referred to as a Node B.

### 2.3.4 UTRAN

UTRAN[2] is a term that includes Node Bs and RNCs in the UMTS network, Figure 2.4. Its main function is to allow connectivity between UEs and the CN.

One RNS can have several Node Bs and there can be many RNSs in a UTRAN. The RNS communicates with Node Bs through a interface called Iub and they can communicate with other RNSs using the Iur interface. Outwards the UTRAN communicates

with the UEs using Uu interface and with CN using Iu interface. Further reading on the interfaces is referred to [3].

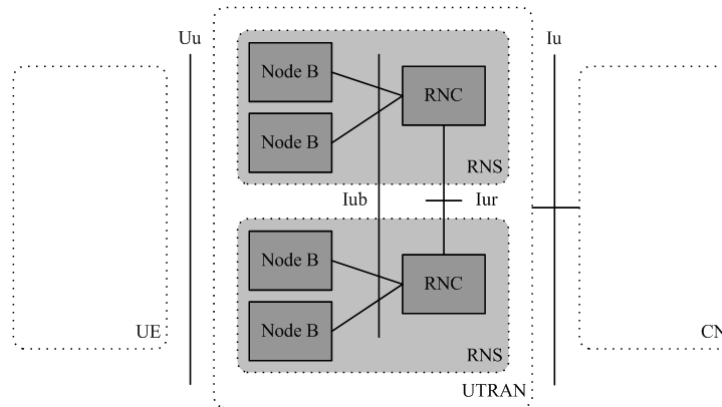


Figure 2.4: UTRAN

The general protocol model for UTRAN interfaces is shown in Figure 2.5. This structure is based on the principle that the layers and planes are logically independent of each other. Therefore, parts of the protocol can be altered or added independently.

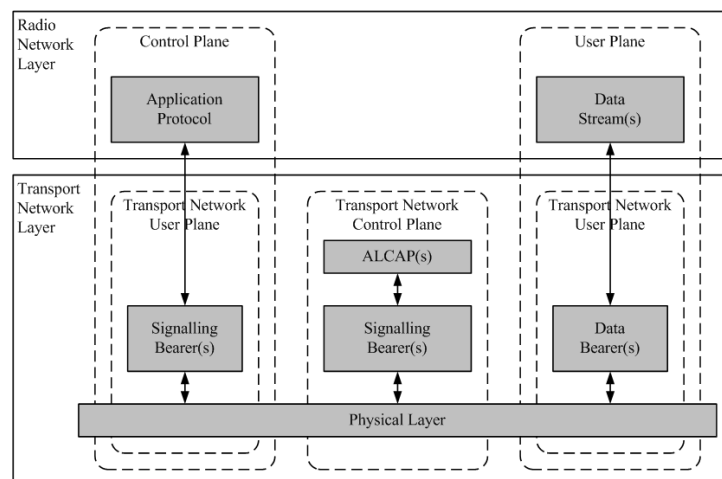


Figure 2.5: UTRAN protocol model

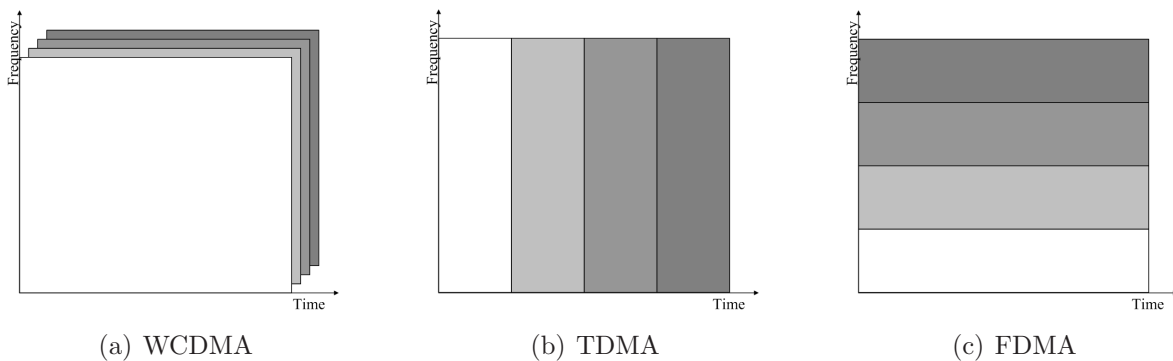
The radio network layer handles all UTRAN related issues, the transport network layer handles the standard transport technology that is to be used. The control plane is used for all UMTS specific control signaling and includes the application protocol and signal

bearers for transporting application protocol messages. The user plane is used for sending and receiving information to the user. It includes the data stream and data bearer that transports the data stream messages. The transport network control plane is used for control signaling within the transport network layer and acts between the control and user plane. It includes the Access Link Control Application Part (ALCAP) with a signal bearer that is necessary for setting up data bearers for the user plane. With the transport network control plane it is possible to make the application protocol in the radio network layer completely independent of the technology selected for the data bearer in the user plane. Transport network user plane only includes the data bearers in the user plane and the signalling bearers for the application protocol.

## 2.4 WCDMA

### 2.4.1 General

WCDMA is the air interface for UMTS. It uses so called spreading codes to allow multiple transmitters to send information simultaneously over one single communication channel at the same frequencies. In Figure 2.6 WCDMA can be seen next to TDMA, which divides transmitters in time slots and FDMA, which divides transmitters in frequency bands.



*Figure 2.6: Air interfaces*

WCDMA is a Direct-Sequence Code Division Multiple Access (DSSSS) system which means that the transmitted information bits are spread over a wider bandwidth than the actual data symbols. Spreading is done by multiplying the data symbols with quasi-random bits called chips. The chips are included in spreading codes that can be of different lengths.

The spreading codes gives the user privacy, since the right spreading code is needed to be able to despread the signal. It also suppresses the narrowband interference. When

the received signal is despread all the interfering signals will be spread. This makes the interference smaller over a wider bandwidth, Figure 2.7.

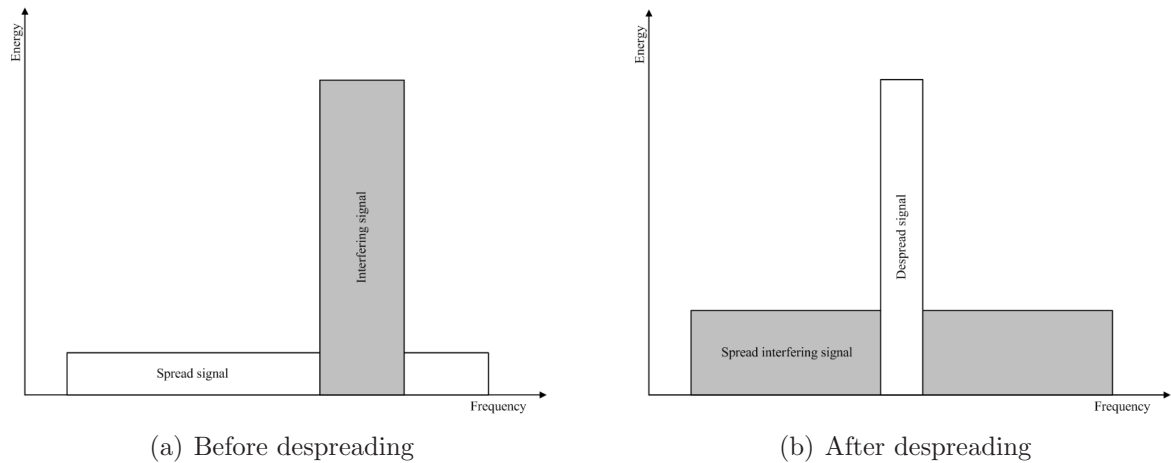


Figure 2.7: Narrowband interference resistance

In WCDMA the chip rate is 3.84 Mcps/s and the channel bandwidth is 5 MHz, Figure 2.8. The carrier spacing can vary between 4.2 MHz and 5.4 MHz in steps of 200 kHz. The selected value is depending on the interference scenario.

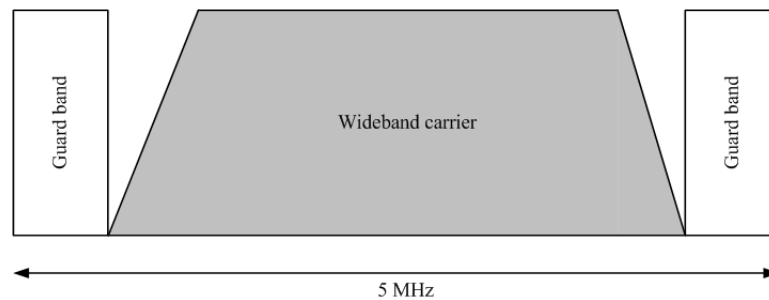


Figure 2.8: WCDMA carrier

Every user in a WCDMA system transmits in 10 ms radio frames, in which the data rate is kept constant. Every frame is further divided into 15 slots. There are two basic modes of operation for WCDMA, Frequency Division Duplex (FDD) and Time Division Duplex (TDD). The first one separates uplink and downlink by frequency and the second one in time slots. UMTS uses FDD mode. As mentioned earlier it has two 5 MHz channels, one for uplink and one for downlink. Another important feature in the WCDMA technology

is handovers, more on this can be read in Section 2.4.5.

The power of a WCDMA signal can be under the noise floor and still be found. This makes it hard to jam the signal since it will just look like noise. The carrier-to-interference ratio C/I determines how strong the signal has to be for detection. For a typical speech service C/I is about -20 dB which means that the signal can be 20 dB under the noise floor. This can be compared with GSM where the C/I is about 11 dB [3].

For WCDMA to work properly, adaptive power control is needed. This minimizes the interference in the system and is based on a Signal-to-Interference Ratio (SIR) value.

## 2.4.2 Spreading

There exists two types of spreading, synchronous and asynchronous. On the uplink asynchronous spreading is used to separate between UEs and synchronous spreading to separate between Physical Channels (PhCHs) on that UE. On the downlink asynchronous spreading is used too separate between the cells in a Node B and synchronous spreading to separate between the UEs in that cell.

Spreading is done by first modulation map the data symbols to 1s and -1s, then multiplying them with the spreading code. When despreading a signal, it is first multiplied with the spreading code and then integrated bit by bit. The spreading and despreading of a data stream with SF 4 can be seen in Figure 2.9. It also shows what happens when despreading with an incorrect spreading code.

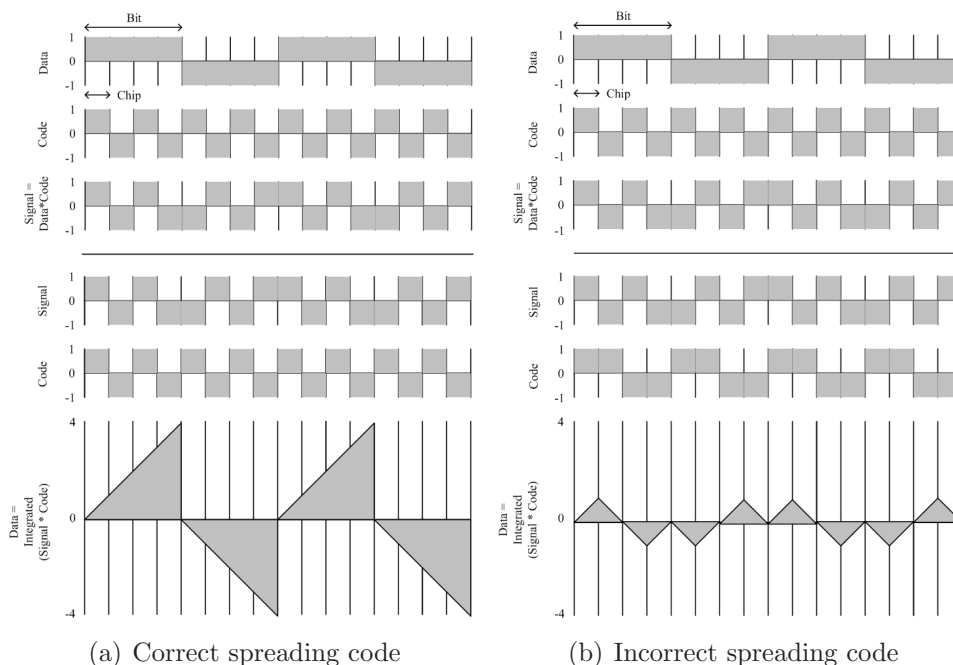


Figure 2.9: Spreading and despreading,  $SF = 4$

### 2.4.2.1 Synchronous

Synchronous mode exploits mathematical properties of orthogonality between bit vectors representing the data. For two vectors to be orthogonal their dot product has to be zero. The spreading codes in WCDMA have these properties. They are so called Orthogonal Variable Spreading Factor (OVSF) Walsh codes, taken from the Walsh tree in Figure 2.10.

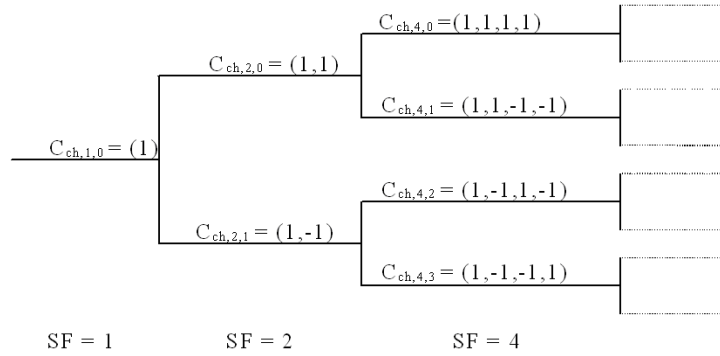


Figure 2.10: Walsh tree

All Walsh codes for the specific Spreading Factor (SF) can be calculated using the recursive formula in Equation 2.1.

$$\begin{aligned}
 C_{ch,1,0} &= 1, \\
 \begin{bmatrix} C_{ch,2,0} \\ C_{ch,2,1} \end{bmatrix} &= \begin{bmatrix} C_{ch,1,0} & C_{ch,1,0} \\ C_{ch,1,0} & -C_{ch,1,0} \end{bmatrix}, \\
 \begin{bmatrix} C_{ch,2^{n+1},0} \\ C_{ch,2^{n+1},1} \\ C_{ch,2^{n+1},2} \\ C_{ch,2^{n+1},3} \\ \vdots \\ C_{ch,2^{n+1},2^{n+1}-2} \\ C_{ch,2^{n+1},2^{n+1}-1} \end{bmatrix} &= \begin{bmatrix} C_{ch,2^n,0} & C_{ch,2^n,0} \\ C_{ch,2^n,0} & -C_{ch,2^n,0} \\ C_{ch,2^n,1} & C_{ch,2^n,1} \\ C_{ch,2^n,1} & -C_{ch,2^n,1} \\ \vdots & \vdots \\ C_{ch,2^n,2^{n-1}} & C_{ch,2^n,2^{n-1}} \\ C_{ch,2^n,2^{n-1}} & -C_{ch,2^n,2^{n-1}} \end{bmatrix} \quad (2.1)
 \end{aligned}$$

For OVSFs to hold their orthogonality two criteria must be fulfilled.

- No other channelization codes on the path from the one used to the root of the tree, or any code belonging to the sub-tree generated from that code can be used.
- The codes must be time synchronized.

In an ideal environment orthogonal codes have a cross-correlation equal to zero. In other words, they do not interfere with each other at all. When noise is present there will always be some kind of interference and the cross-correlation will differ from zero.

With longer spreading codes the cross-correlation properties will be better and more transmitters will be able to send at the same time. This also increases the spreading of the signal and will thereby decrease the bandwidth.

### 2.4.2.2 Asynchronous

When time synchronization between transmitters is impossible to achieve, asynchronous spreading mode can be used. Instead of orthogonal spreading codes it uses Gold codes<sup>3</sup> created in a Gold code generator. These sequences are statistically uncorrelated and the sum of a large number of Gold codes results in Multiple Access Interference (MAI), that is approximated by a Gaussian noise process. If all receivers are reached with signals of the same power level, the noise power of the MAI increases in direct proportion to the number of transmitters. With too many transmitters in the system the signal energy will rise to a level so high that the receiver will not be able to distinguish them from each other anymore. An example of signal despreading can be seen in Figure 2.11.

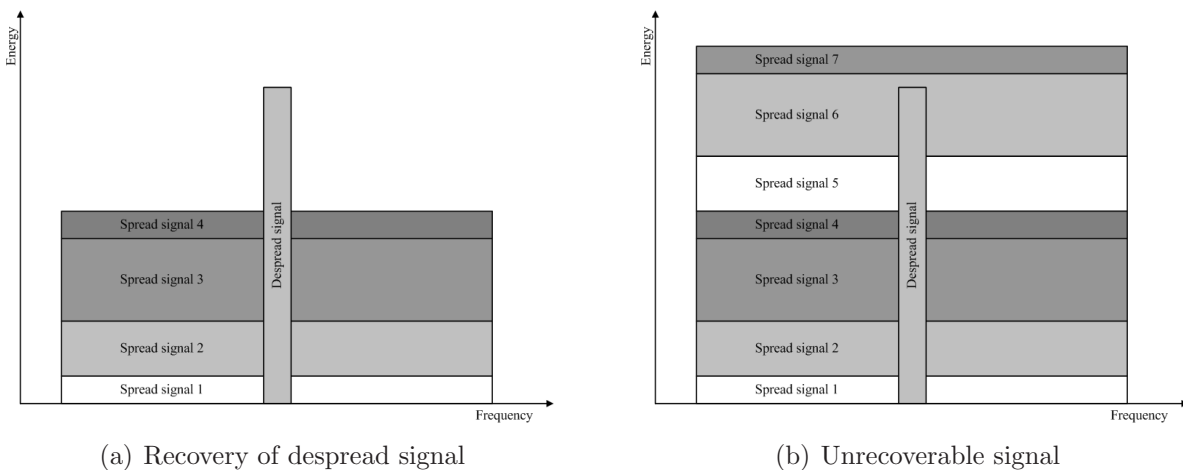


Figure 2.11: Despreading

To get data from a transmitter, the receiver has to correlate the received signal with

<sup>3</sup>A set of binary sequences.

the right spreading code. If the spreading code exists in the signal, a correlation peak will be found when the signal has been time synchronized. The data is then retrieved by despreading the signal with the spreading code.

One very important property with asynchronous WCDMA is the transmitting power of every user. Since the rejection of unwanted signals is only partial, one user could overwhelm all others by transmitting with too much power. Therefore, power control is a very important feature.

### 2.4.3 Multipath and rake receivers

When sending radio signals over land there are high probability that one or more obstacles will be in the way. These can cause multiple reflections, diffraction and attenuation of the radio signals. This is called multipath propagation, Figure 2.12. Two major effects can result from this.

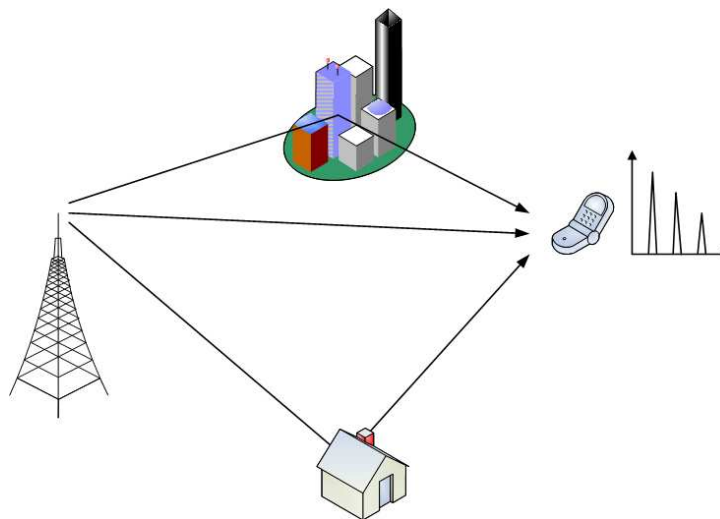
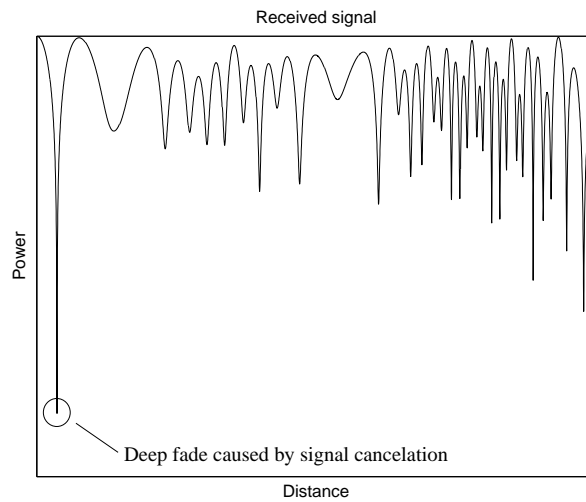


Figure 2.12: Multipath propagation

InterSymbol Interference (ISI), this is when the same signal arrives at different time instances to the receiver with at least one chip offset ( $0.26 \mu\text{s}$  at  $3.84 \text{ Mcps/s}$ ) between the signals and will cause the chips to interfere with each other. The pattern for this can be quite easily distinguished for different areas. The delay may vary from  $1 \mu\text{s}$  in urban areas to  $20 \mu\text{s}$  in hilly areas.

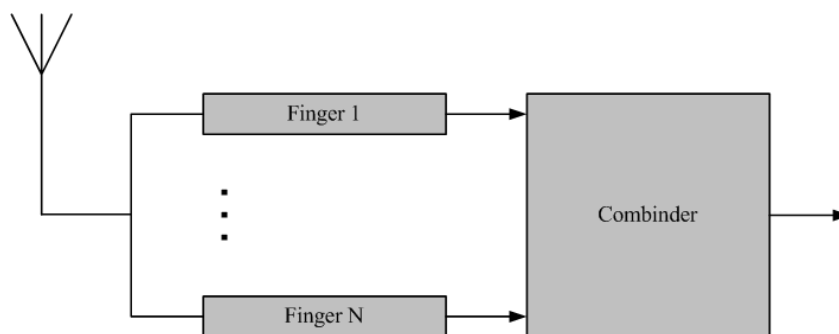
Fast fading, this happens when the same signal is received multiple times with paths very close to each other. Then there will be phase offsets in the carrier wave and these can cause signal cancellation due to destructive summation of the received signals, Figure 2.13. Fast fading is related to the carrier wave frequency, multipath geometry and the

velocity of a moving receiver. As a rule of thumb there can be up to four fades/s per km/h for a moving receiver.



*Figure 2.13: Fast fading*

Multipath diversity can be achieved by using RAKE receivers<sup>4</sup>, Figure 2.14. These have multiple fingers (correlators) which each are time delayed to a positions where multipath reflections are expected. The signals in every finger are demodulated and then phase adjusted to compensate for the fast fading. All fingers are then combined to one signal using equal-gain combining, Maximal-Ratio Combining (MRC) or strongest signal selection.



*Figure 2.14: Rake receiver*

<sup>4</sup>It is called a RAKE receiver because it reminds of a garden rake in both appearance and functionality.

Fast power control, strong coding, interleaving and retransmissions further adds redundancy and time diversity to the signals which helps the receiver to recover bits across fading.

#### 2.4.4 Near/far problem and power control

Power control is one of the most important features in WCDMA, especially on the uplink. Without it, a single overpowered UE could block a whole cell. It is also important in the downlink, this enables the Node B to reach all the UEs in its cells with sufficient power. When a UE is close to the cell edge and no power control is used it could very easily be overshoot by a UE closer to the Node B, this is called the near/far problem can be seen in Figure 2.15.

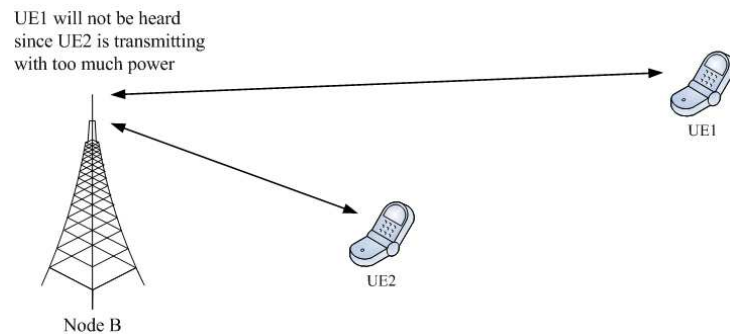


Figure 2.15: Near/far problem

There are three types of power control. Closed-loop, outer-loop and open-loop. The open-loop is basically used to provide a coarse initial power setting at the beginning of a connection.

The main power control is the closed-loop. On the uplink, it compares a SIR value with a defined SIR value that is determined by the RNC. Depending on if the measured value is higher or lower than the expected value the base station will command the UE to either lower or increase its power. This will happen at a frequency of 1500 Hz (on a slot basis). The same technique is used on the downlink, here the near/far problem does not exist since there is only one base station transmitting. Instead the power is managed so that it reaches the UEs at the cell edge with sufficient power, to sustain other-cell interference. This is also a method of enhancing weak signals caused by fast fading due to insufficient error correction, like when sending data over a low speed channel.

The outer-loop power control is controlling the SIR value depending on the needs of individual radio links. This is controlled to optimizing power consumption in the UEs, so that it does not send a stronger signal than necessary. It is typically implemented in

some sort of reliability indicator at the uplink.

Figure 2.16 shows how all three power controls are used when setting up an uplink Dedicated Physical Channel (DPCH).

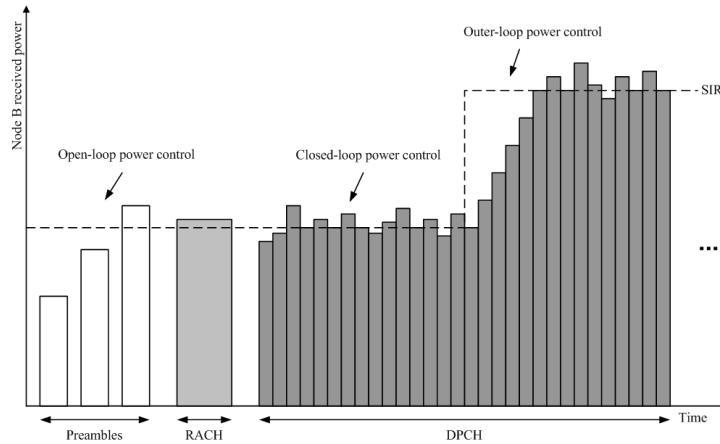


Figure 2.16: Power control when setting up an uplink DPCH

## 2.4.5 Handovers

Handovers are when UEs are moved from one cell to another while still maintaining their connection. This is what makes it possible for the UEs to be mobile. If a UE comes out of range of a cell and in range of another cell these two will make a handover and the UE will be transferred to the new cell. There are four different types of handovers.

- Softer handover
- Soft handover
- Inter-frequency
- Inter-system

Softer handover is when the UE changes between two different cells in the same Node B. Soft handover is when the UE changes between two Node Bs. Inter-frequency is when the UE changes between two frequencies but still using the same technology and inter-system is when the UE changes between two different technologies.

When doing a inter-frequency handover something called compressed mode is used. This basically means that either the SF is decreased or bits are punctured to allow empty time slots for inter-frequency measurements.

## 2.5 Physical layer

This section will describe the physical layer at the UEs side of the network on the uplink.

### 2.5.1 General

The physical layer (layer 1) is the lowest layer in the WCDMA radio interface protocol architecture, Figure 2.17. Here logical data from higher layers are turned into analog signals and sent out in the air. The physical layer is required for support of variable bit rates at higher layers and for multiplexing of several services to one connection. Above the physical layer, on layer 2 is the Medium Access Control (MAC) and under the physical layer is the radio interface. The physical layer is controlled by the Radio Resource Control (RRC), that is placed in layer 3. Measurements reported to higher layers[4] are not covered in this master's thesis.

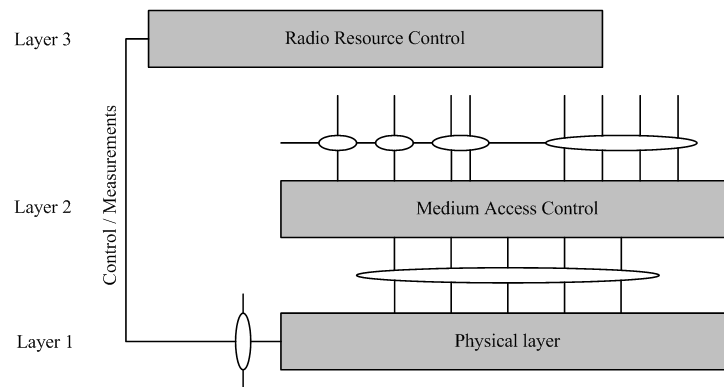


Figure 2.17: WCDMA radio interface protocol architecture around the physical layer

Functions given by the physical layer are listed below:

- Forward Error Correction (FEC) encoding/decoding.
- Radio measurements and indications to higher layers.
- Macrodiversity distribution/combining and soft handover execution.
- Error detection.
- Multiplexing and demultiplexing of Transport Channel (TrCH)/Coded Composite Transport Channel (CCTrCH).
- Rate matching.

- Modulation/demodulation, spreading/despreading of PhCH.
- Mapping of CCTrCHs on physical channels
- Frequency and time synchronization.
- Closed-loop power control.
- Power weighting and combination of physical channels.
- Radio Frequency processing.

## 2.5.2 Medium Access Control

The MAC maps logical channels onto transport channels. It is the lowest sublayer in layer 2 and consists of three logical entities, Figure 2.18. MAC-b, that handles the broadcast channel, MAC-c/sh, that deals with the common and shared channels and MAC-d, that handles the dedicated channel.

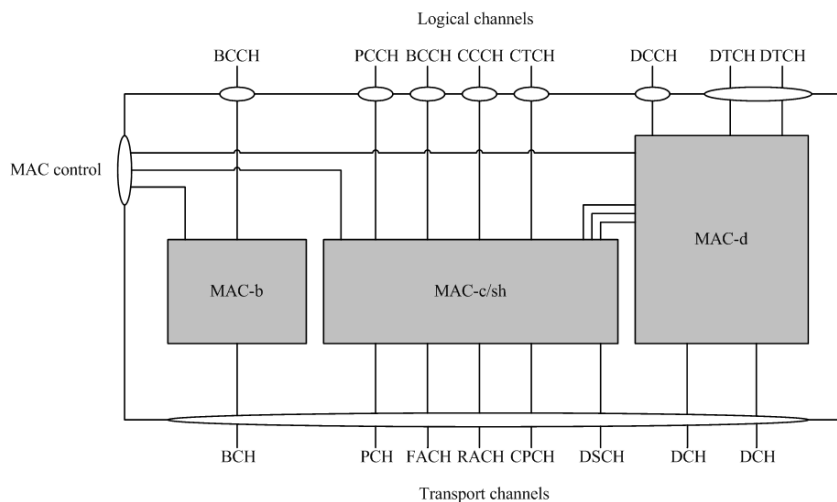


Figure 2.18: MAC sublayer

## 2.5.3 Radio Resource Control

The RRC controls the configuration of the lower layers in the UTRAN protocol stack. It consists of three services: general control, notification and dedicated control.

General control is an information broadcast service that broadcast the information to all UEs within a certain area.

Notification includes paging and notification broadcast service. The paging is broadcasting paging information in a certain area, but it addressed to a specific UE or UEs.

The notification broadcast service is defined to provide information broadcast to all UEs in a cell or cells.

Dedicated control includes the establishment and release of a connection and the transfer of messages using this connection. These connections can be both point-to-point or group connections.

### 2.5.4 Synchronization

UTRAN needs synchronization to send and receive information. To get this, synchronization parameters[5] are used. The ones relevant to this master's thesis can be seen in the list below.

- Node B Frame Number (BFN)
  - Counts number of radio frames.
- System Frame Number (SFN)
  - Is sent on Broadcast Channel (BCH) and is used paging groups and system information scheduling etc. SFN is BFN adjusted with  $T_{cell}$ .
- Connection Frame Number (CFN)
  - CFN is the frame counter used for the layer 2/transport channel synchronization between UE and UTRAN.
- Frame offset
  - Frame offset is a radio link specific layer 1 parameter used to map the CFN, used in the transport channel, into the SFN that defines the specific radio frame for the transmission on the air interface.
- $T_{cell}$ 
  - $T_{cell}$  represents the timing delay used for defining the start of Synchronization Channel (SCH), Common Pilot Channel (CPICH) and the Downlink Scrambling Code(s) in a cell relative BFN.

### 2.5.5 Transport blocks

Data from the MAC is put in Transport Block Sets (TBSs)[6], these are sent to the physical layer in a Frame Protocol (FP) every Time Transmission Interval (TTI). The FP structure can be seen in Appendix B. The TTI length depends on the transport channel and can be 10, 20, 40 or 80 ms long. Data in a TBS is defined by the Transport Format (TF). This consists of two parts, semi-static and dynamic. The semi-static part is common to all transport formats and consists of:

- TTI
- Error protection scheme
- Size of Cyclic Redundancy Check (CRC)
- Static rate matching parameter

The dynamic part can change with the transport format and consists of:

- Transport Block (TB) size
- TBS size

## 2.5.6 Radio frames

Data outputted from the physical layer is assembled in 10 ms radio frames, that each consist of 38400 chips. These frames are further divided into 15 slots holding 2560 chips. The physical frame structure differs with different transport channels.

## 2.5.7 Channels

There are three types of channels in UTRAN. Logical channels, that defines the data transfer service offered by layers above the MAC. Transport channels, that defines how and with what type of characteristics the data is to be transferred by the physical layer. Physical channels, that defines the exact physical characteristics of the radio channels.

These three types of channels can either be dedicated, common or shared. The difference between them is that the common and shared channels are shared by multiple users in a cell. While the dedicated channels are reserved for one user only.

The transport channels each have a Transport Format Indicator (TFI). This is sent at the same time as the data and is collected in the Transport Format Combination Indicator (TFCI) at the physical layer, Figure 2.19. This way, the physical layer will have control over which transport channels are active for the current frame. There are cases when no TFCI is needed, this is when the channel has a fixed data rate. All physical channels are multiplexed into a CCTrCH.

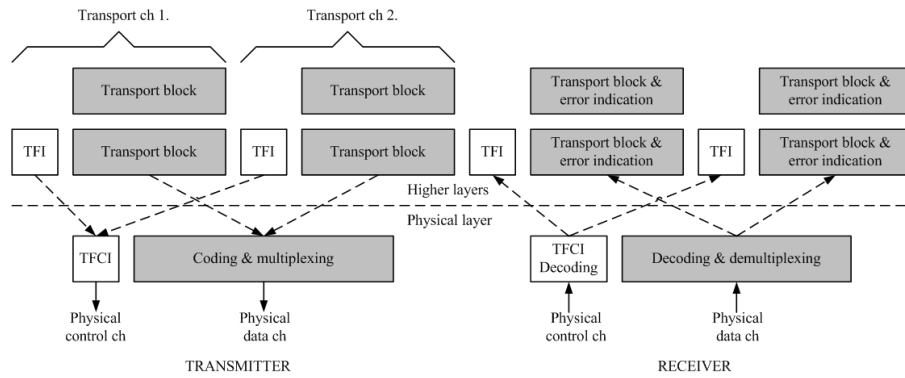


Figure 2.19: Interface between MAC and the physical layer

The mapping between transport channels and physical channels[7] can be seen in Figure 2.20. In this master thesis the transport channel Dedicated Channel (DCH) and its physical channels Dedicated Physical Data Channel (DPDCH) and Dedicated Physical Control Channel (DPCCH) are the ones that will be used.

Transport Channels	Physical Channels
DCH	Dedicated Physical Data Channel (DPDCH) Dedicated Physical Control Channel (DPCCH)
E-DCH	Fractional Dedicated Physical Channel (F-DPCH) E-DCH Dedicated Physical Data Channel (E-DPDCH) E-DCH Dedicated Physical Control Channel (E-DPCCH) E-DCH Absolute Grant Channel (E-AGCH) E-DCH Relative Grant Channel (E-RGCH) E-DCH Hybrid ARQ Indicator Channel (E-HICH)
RACH	Physical Random Access Channel (PRACH)
BCH	Common Pilot Channel (CPICH)
FACH	Primary Common Control Physical Channel (P-CCPCH)
PCH	Secondary Common Control Physical Channel (S-CCPCH)
	Synchronisation Channel (SCH) Acquisition Indicator Channel (AICH) Paging Indicator Channel (PICH) MBMS Notification Indicator Channel (MICH)
HS-DSCH	High Speed Physical Downlink Shared Channel (HS-PDSCH) HS-DSCH-related Shared Control Channel (HS-SCCH) Dedicated Physical Control Channel (uplink) for HS-DSCH (HS-DPCCH)

Figure 2.20: Mapping between transport and physical channels

### 2.5.7.1 DCH

DCH is a transport channel that is used to send data to one UE only. There can be several DCHs active at the same time and these can have different services. DCH is mapped on to two physical channels, DPDCH and DPCCH. The uplink frame structure of these PhCHs can be seen in Figure 2.21. DPDCH and DPCCH are always frame aligned with each other.

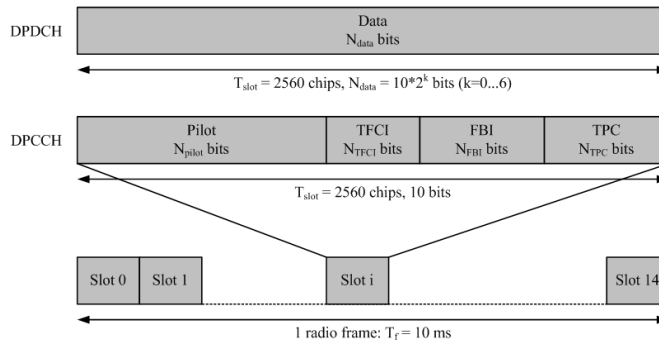


Figure 2.21: Uplink DPCH radio frame structure

### 2.5.7.2 DPDCH

DPDCH is the physical channel that carries the DCH. There can be up to six DPDCH active at the same time on one physical link. These PhCHs will be evenly divided on the  $I^5$  and  $Q^6$  branches in the Hybrid Phase Shift Keying (HPSK) modulation. DPDCH is spread with a SF between 4 and 256.

### 2.5.7.3 DPCCH

DPCCH is used to carry control information for DCH, generated at the physical layer. One radio frame consists of 15 DPCCH slots that each holds 3-8 known pilot bits to support channel estimation for coherent detection, 2 Transmit Power-Control (TPC) command bits, 0-1 bits of Feedback Information (FBI) and an optional TFCI that consists of 0-4 bits. TFCI informs the receiver about the instantaneous transport format combination of the transport channels mapped to the simultaneously transmitted uplink DPDCH radio frame. There is one and only one DPCCH on each radio link. The pilot bits, TPC, FBI are unique for every slot while the TFCI bits are mapped between all 15 slots in a radio frame using (32, 10) sub-code of second order Reed-Muller code<sup>7</sup>.

<sup>5</sup>In-phase.

<sup>6</sup>Quadrature-phase.

<sup>7</sup>A linear error-correcting code.

DPCCH is always multiplexed on the Q branch and spread with SF 256.

## 2.5.8 Channel coding and multiplexing for uplink

This is a summary from the 3GPP specification [8]. Some of the coding parts are further described in Appendix A. The coding and multiplexing flow chart can be seen in Figure 2.22.

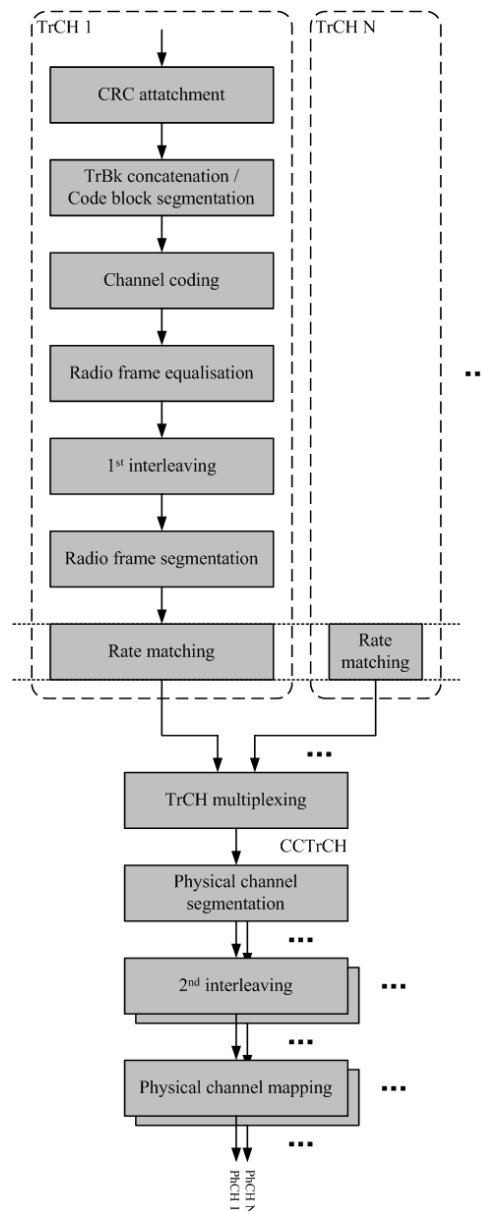


Figure 2.22: Transport channel multiplexing structure for uplink

### 2.5.8.1 CRC attachment

The CRC is calculated on a TBS received from the MAC and added at the end of each TB in that set. It is used for error detection at the receiving side and to calculate the Bit Error Ratio (BER). The CRC can be of five different lengths, 24, 16, 12, 8 and 0 bits. This is determined by higher layers. Information on how the CRC calculation is performed can be read in Appendix A.

### 2.5.8.2 Transport block concatenation / code block segmentation

After CRC attachment the TBS is concatenated. This means that if there are more than one TB in the TBS they will be put together in a serial manner.

After the block concatenation the TBS are put in block segments of equal sizes. This is done to optimize the algorithms for channel coding and will only be done if the TBS size is greater than  $Z$ , where  $Z$  is 504 bits for convolutional coding and 5114 bits for turbo coding. The number of code blocks  $C$  is determined by Equation 2.2, where  $X$  is the total number of bits.

$$C = \left\lceil \frac{X}{Z} \right\rceil \quad (2.2)$$

If  $X$  is not a multiple of  $Z$  then filler bits are added to the beginning of the block sequence to get an even amount of blocks. The filler bits are always 0s.

### 2.5.8.3 Channel coding

When the TBs have been segmented they are encoded with a channel coding. The channel coding is a FEC coding that is applied to enable the receiving side to correct errors. Two types of channel codings are used in UMTS, convolutional coding and turbo coding.

After encoding, the number of output bits in each code block will be  $Y_i$ , as described in the list below.  $K_i$  is the number of bits into the encoder.

- Convolutional coding
  - Rate 1/2:  $Y_i = 2K_i + 16$ .
  - Rate 1/3:  $Y_i = 3K_i + 24$ .
- Turbo coding
  - Rate 1/3:  $Y_i = 3K_i + 12$ .

The coding that will be used depends on which transport channel the TBs comes from, Table 2.1.

Type of TrCH	Coding scheme	Coding rate
BCH	Convolutional coding	1/2
PCH		
RACH		1/3, 1/2
DCH, FACH	Turbo coding	1/3

Table 2.1: Coding types for different transport channels.

More on FEC codings can be read in Appendix A.

#### 2.5.8.4 Radio frame equalization

This is padding of the input bit sequence to ensure that the output can be segmented into an even number of radio frame segments with the same size. It is only performed on the uplink and depends on the TTI of the channel.

#### 2.5.8.5 1<sup>st</sup> interleaving

The 1<sup>st</sup> interleaving is using a block interleaver with inter-column permutations. The incoming bit sequence will fill a matrix row by row. The number of columns  $C$  in the matrix are decided from Table 2.2 and the number of rows  $R$  are calculated<sup>8</sup> using Equation 2.3 where  $X$  is the number of input bits.

$$R = \frac{X}{C} \quad (2.3)$$

When a matrix is created, the columns are shifted according to the inter-column permutation patterns in Table 2.2. The interleaved bit sequence is outputted column by column.

TTI	Number of columns	Inter-column permutation patterns
10 ms	1	<0>
20 ms	2	<0, 1>
40 ms	4	<0, 2, 1, 3>
80 ms	8	<0, 4, 2, 6, 1, 5, 3, 7>

Table 2.2: Inter-column permutation patterns for 1<sup>st</sup> interleaving.

<sup>8</sup>Radio frame equalization guarantees that the number of input bits is an integer multiple of the number of radio frames in the TTI.

More information on interleaving can be read in Appendix A.

### 2.5.8.6 Radio frame segmentation

If the TTI is longer than 10 ms the input bit sequence will be segmented into a number of consecutive radio frames. This can be done since the radio frame equalization has ensured that the bit sequence is a multiple of numbers of radio frames in the TTI.

### 2.5.8.7 Rate matching

The bit rate on transport channels can change every TTI. To match this with the bit rate on the physical channels, rate matching is applied. The rate matching can either repeat bits if there are too few, or puncture bits if there are too many.

The goal is to get the data from all active TrCHs on that TTI to completely fill the radio frame. This is necessary since the radio frames<sup>9</sup> that are sent in the air always have to be filled.

The semi-static rate matching attribute RM is used when the number of bits to be repeated or punctured is calculated. This can only be changed at higher layers. On the uplink the rate matching also determines the SF and number of PhCH needed.

### 2.5.8.8 TrCH multiplexing

Every 10 ms, one radio frame from each transport channel is delivered to the TrCH multiplexing. These frames are serially multiplexed into a CCTrCH, which will hold exactly the number of bits that shall fill one radio frame that is to be sent in the air.

### 2.5.8.9 Physical channel segmentations

When more than one physical channel is used, the physical channel segmentation divides the bits among the physical channels.

### 2.5.8.10 2<sup>nd</sup> interleaving

The 2<sup>nd</sup> interleaving is just like the 1<sup>st</sup> one, a block interleaver. Here the number of columns are predefined to 30 and the rows are calculated. If the matrix does not get filled zero padding is used. The columns are rearranged according to the permutation pattern in Table 2.3. If padding was performed, these bits will be pruned as a final step.

---

<sup>9</sup>38400/SF bits.

Number of columns	Inter-column permutation patterns
30	<0, 20, 10, 5, 15, 25, 3, 13, 23, 8, 18, 28, 1, 11, 21, 6, 16, 26, 4, 14, 24, 19, 9, 29, 12, 2, 7, 22, 27, 17>

Table 2.3: Inter-column permutation patterns for 2<sup>nd</sup> interleaving.

### 2.5.8.11 Physical channel mapping

Here the transport channels are mapped on the physical channels. On the uplink, the radio frames are either completely filled or not used at all. Only exception is when the UE is in compressed mode. The transmission can then be turned off during consecutive slots of the radio frame.

## 2.5.9 Modulation

The modulation scheme used by the uplink is HPSK.

### 2.5.9.1 Modulation mapping

To be able to perform further coding the bits represented by 1s and 0s has to be converted to real values. The 1s will take the value -1 and the 0s will take the value 1. This makes it possible to perform addition with the signals when they are converted to voltages in the mixing procedure, Section 2.5.12.

### 2.5.9.2 HPSK

HPSK is a complex scrambling scheme that multiplexes the PhCHs on I and Q branches, where the Q branch is multiplied with the complex number  $j$ . HPSK uses four phases and can decode one I/Q value per symbol. The constellation diagram for HPSK is shown in Figure 2.23. It can be seen that I and Q branches are using unequal power levels. This is the case on the uplink and is controlled by the gain factor  $\beta$ .

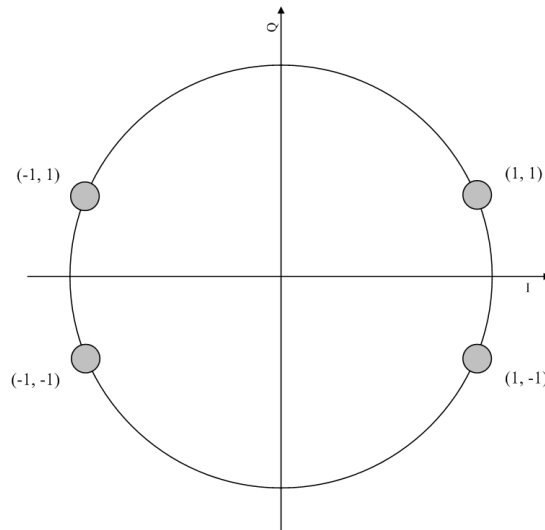


Figure 2.23: Constellation diagram for HPSK

The reason for using HPSK is that it decreases the number of zero crossings<sup>10</sup> in the output by only allowing specific channelization codes. This improves the peak-to-average power ratio of the signal and reduces the power consumption on the uplink power amplifier. This is very important since the uplink power amplifier is the biggest power consumer in the UE.

## 2.5.10 Spreading

### 2.5.10.1 Channelization

Channelization[9] is used to separate different physical channels from the same transmitter. This is done with synchronous spreading and the use of orthogonal spreading codes, taken from the Walsh tree. The lengths of these codes can vary between 4 and 256 chips and are assigned on a per-UE basis on the uplink.

### 2.5.10.2 Scrambling

Scrambling[9] is used to separate different transmitters from each other. This is done with asynchronous spreading and the use of Gold codes.

The scrambling code is a complex valued sequence divided in two categories. Long codes, that consists of 38400 chips and short codes that consists of 256 chips. Both are generated in scrambling code generators, Figure 2.24. The uplink uses both codes.

All transmitters have a unique 24 bit scrambling sequence number  $n$  that is loaded in the scrambling code generator. This gives  $2^{24}$  long and  $2^{24}$  short scrambling codes which

<sup>10</sup>Zero crossing is when the transition has to go via origo in the constellation diagram.

all have good auto-correlation properties.

The scrambling lies on top of the channelization and does not change the bandwidth of the signal, since the length of the scrambling code is equal to the length of a radio frame.

Complex scrambling is in relation with HPSK a way to improve the peak-to-power ratio of the signal and reduce the power consumption on the uplink power amplifier. The scrambling code generated is a so called Walsh rotator sequence. This sequence guarantees that every other phase shift will be exactly  $90^\circ$  between two sequent scrambled values, which will further halve the number of zero crossings in the HPSK constellation diagram, Figure 2.23.

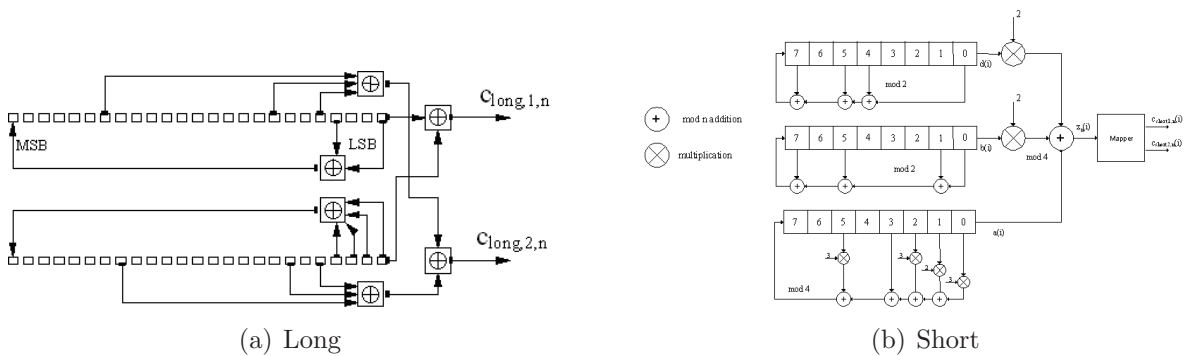


Figure 2.24: Scrambling sequence generators

### 2.5.11 Channel multiplexing and I/Q mapping

When all active TrCHs have been spread with a channelization code and multiplied with gain factor  $\beta$  they will be multiplexed together and I/Q mapped[9].  $\beta$  is used to adjust the power between the PhCHs. The I/Q mapping is done to put the data and control channel on I and Q branches. These are  $90^\circ$  phase shifted and makes it possible to send them in parallel. If more than one data channel is used these will also be divided on the branches. I/Q mapping is simply done by multiplying the channels on the Q branch with the complex number  $j$ . In Figure 2.25 multiplexing and I/Q mapping for DPCHs is performed. Six data channels and one control channel is the maximum that can be used for DCH. After channel multiplexing and I/Q mapping the I and Q branches will be scrambled and forward to the mixing.

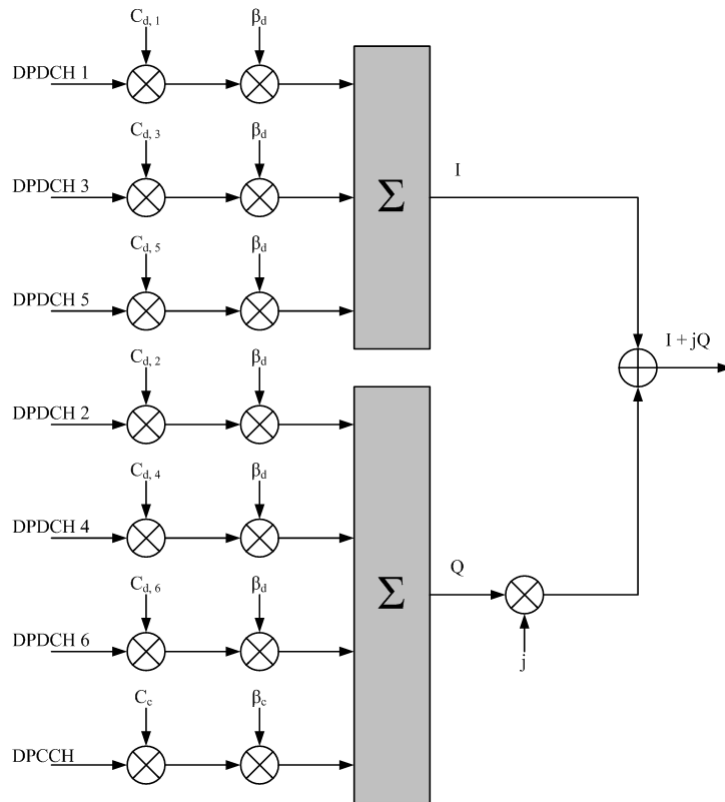


Figure 2.25: Channel multiplexing

### 2.5.12 Mixing

The mixing procedure can be seen in Figure 2.26. I and Q branches are separated, pulse shaped with a raised-cosine filter, multiplied with a carrier wave, summed together and sent out in the air.

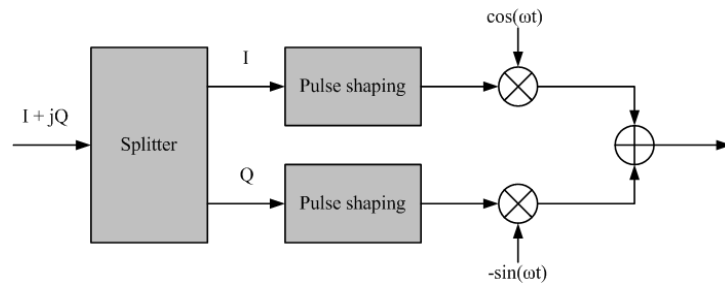


Figure 2.26: Mixing

---

## CHAPTER 3

---

# Test Environment

### 3.1 TXAD3

TX Adapter Card 3 (TXAD3) is a platform developed at Ericsson and used to verify functionality on the Uplink and Downlink. As this is Ericsson proprietary it will not be described in detail, only things relevant to understand the implementation in this master's thesis will be discussed. TXAD3 consists of Digital Signal Processors (DSPs), Field-Programmable Gate Array (FPGAs), a Board Processor (BP) and Synchronous Dynamic Random Access Memory (SDRAM).

#### 3.1.1 DSP

The DSPs are of the model TMS320C6416T[10] from Texas Instruments. These are very powerful fixed-point DSPs running at a clock frequency of 850 MHz. The DSPs have two External Memory Interfaces (EMIFs)[11], EMIFA is used to access the external DDR memory and EMIFB is connected to a FPGA. To send data between DSP and EMIF, Quick Direct Memory Access (QDMA)[12] is used.

#### 3.1.2 FPGA

There are three FPGAs on TXAD3, These are of the type Virtex 5[13] from Xilinx. The FPGAs are the interface between the DSPs and the antenna stimuli bus.

### 3.1.3 Operating system

The Operating System (OS) running on the TXAD3 is Operating System Embedded (OSE)<sup>1</sup>. On the DSPs a lightweight version of OSE called OSE Compact Kernel (OSEck) is used.

---

<sup>1</sup>A Real-Time Operating System (RTOS) developed by ENEA.

---

# CHAPTER 4

---

## Method

### 4.1 Systemization

The real-time test environment designed for TXAD3 can be seen in Figure 4.1. To generate sampled antenna data, all three blocks have to be implemented. This master's thesis only cover implementation of the baseband encoding block.

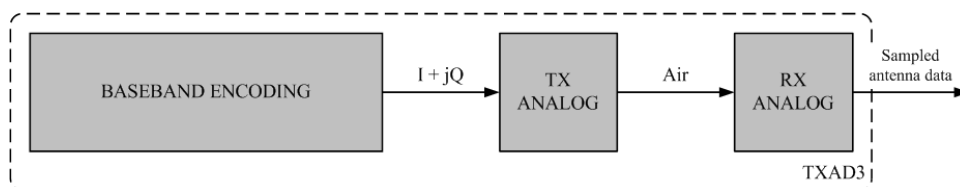


Figure 4.1: Real-time test environment on TXAD3

#### 4.1.1 Baseband encoding

Baseband encoding includes the uplink channel coding and multiplexing described in Section 2.5.8. It also does modulation mapping, channelization and scrambling. The output from the baseband encoding is I/Q data.

#### 4.1.2 TX analog

Before sending the baseband encoded data out in the air it has to go through the analog transmitter. This consists of a Digital-to-Analog (D/A) converter that includes

pulse shaping with a Root-Raised Cosine (RRC) filter. After that the signal is mixed up on a carrier wave. In the test environment the mixing is transparent and not actually performed.

### 4.1.3 RX analog

The first part in the analog receiver is a Low Noise Amplifier (LNA). After the LNA the signal is mixed down to baseband and run through a RRC filter to remove unwanted frequencies. Just like with TX analog part the mixing is a transparent step. By using RRC filters both on TX and RX analog parts will create a so called matched filter, this maximizes Signal-to-Noise Ratio (SNR) of the signal. When the signal is filtered it is sampled with a Analog-to-Digital (A/D) converter at  $2 * F_c$  and converted into a quantified value.

## 4.2 Implementation of the baseband encoder

### 4.2.1 MATLAB simulations

The entire encoding scheme was first written in MATLAB. This was to get an understanding of how every step worked in detail. To use MATLAB was a given choice since it is designed to work with matrices which the baseband encoding mostly consists of. Since a lot of bit matrices are used in the encoding, MATLAB was a given choice. It also makes it simple to see the result of every step. One disadvantage is though that MATLAB is quite slow and it takes some time to process long bit matrices. This is probably caused by the fact that all functions implemented are written in a way so that they could easily be converted to C code.

### 4.2.2 Division of functionality

Since no VHDL code for the uplink was written for the FPGAs on TXAD3, the communication between the DSP and FPGA has only been prepared using EMIF and QDMA interfaces on the DSP.

The FPGA has limited amount of memory so it is decided to send burst slots (5120 chips) into a FIFO queue. The scrambling is decided to be done on the FPGA, so a scrambling code generator and a scrambler has been implemented in VHDL and simulated in Modelsim. The implementation of TX and RX analog parts is not a part of this master thesis.

## 4.2.3 Software

### 4.2.3.1 Code structure

The code is divided into modules called `trch`, `cctrch`, and `phch`. These modules each have their own coding module that holds single functions needed for doing its encoding functionality. The super module decides which of the functions is needed in the encoding process. This way it is very easy to have an overview of the system and functionality can be added or modified as simple as they can be removed. Debugging is also very simple to do since data can be extracted from every coding step.

This master's thesis is focused on DCH users, but since all encoding functionality are in individual modules other channels can easily reuse these functions in their own configuration.

### 4.2.3.2 Bit vectors and matrices

To represent bit matrices two structures are used. One is called `bitarr`, it consists of a size field and a data pointer. This makes it simple to create and handle bit arrays of different lengths and send them between functions.

To handle more dimensions, a linked list is used, where every row or column in the linked list is a `bitarr`. The reason for choosing a linked list is that it also can be used for listing other things in the application.

### 4.2.3.3 State variables

The application is using three state variables, they are:

- `trch_state`
- `cctrch_state`
- `phch_state`

The `trch_state` holds all needed parameters for the first part of the encoding, up to rate matching. It also has a buffer for the last encoded TTI. This makes it possible to only have to do encoding every  $F_i$ <sup>1</sup>.

The `cctrch_state` holds the needed parameters for rate matching. It also holds two buffers of the last encoded frame.

The `phch_state` state holds the parameters needed for `phch` encoding. This includes physical channel segmentation, second interleaving, physical channel mapping, conversion to real values, channelization and scrambling. It also has two buffers for radio frame slots, so one can be encoded while the other one is sent.

---

<sup>1</sup>Number of frames in a TTI.

To have data buffers in every state requires more memory, but it simplifies the system and gives a better overview of it. This way everything can be also prepared in advance before having to send a radio frame, as well as it can both send and encode the radio frames at the same time.

#### 4.2.3.4 Users

The users consist of a user id, *trch\_state*, *ctrch\_state*, *phch\_state*, scrambling code polynomial n, active flag, start BFN, rlsr BFN and release BFN.

There are three functions available for the users, these are:

- **setup**
  - Activates a user
  
- **rlsr**
  - Changes RAB for a user
  
- **release**
  - Deactivates a user

The time at which setups, rlsrs and releases occur are predefined in the test case and relative to BFN.

#### 4.2.3.5 Encoding procedure

The application flow chart can be seen in Figure 4.2.

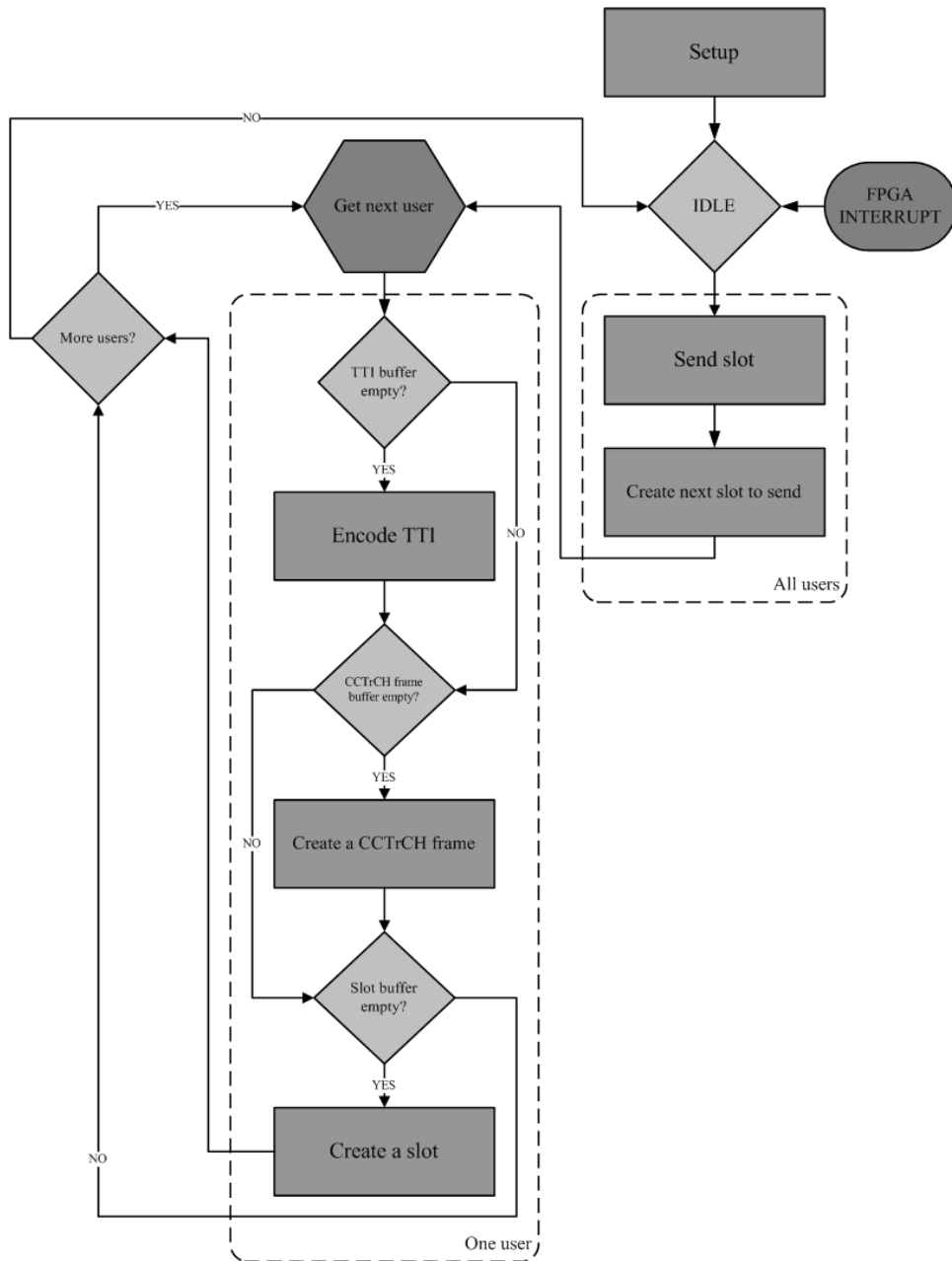


Figure 4.2: Application flow chart

Before the encoding can start the system has to setup all users and RABs in the test case. This information is at the moment hard coded on the DSP, but the thought is that all such parameters shall come from higher layers in the test bench<sup>2</sup>.

The application will start by filling all buffers. Then it will enter idle mode, and wait

<sup>2</sup>Includes all parts needed for layer 1 and layer 2 verification of a Node B.

for an interrupt from the FPGA, meaning that the FPGA needs more data in its First In, First Out (FIFO). When the DSP gets an interrupt from the FPGA it will send one slot of I/Q data. If more than one user is active the data in the slot will be divided among the users. When the slot is sent a new slot will be created, to be sent on the next interrupt.

The application will go through all users and see if any encoding has to be done either on the `trch`, `cctrch` or `phch`. Meaning that any of the data buffers are empty. When all users has been gone through it will enter the idle mode again and wait for an FPGA interrupt.

## 4.3 Channel coding and multiplexing

Flow chart of the baseband encoder can be found in Appendix C.

### 4.3.1 DCH

#### 4.3.1.1 Randomly generated data

The UE data is randomly created in real-time for every TTI. So new unique data will be available in every radio frame.

#### 4.3.1.2 CRC attachment

The CRC is calculated with a lookup table. This means that a precalculated table is used to calculate the CRC one byte at a time instead of one bit at a time, and this speeds it up by 8 times. It uses zero as initial reminder and reflects the CRC reminder at the end. The CRC is attached at the end of every TB and then the TBs are serially concatenated.

The function can calculate CRC of lengths 24, 16, 12 and 8. Memory needed for the lookup tables are  $256 + 256*2 + 256*2 + 256*4 = 2304$  bytes.

To make the CRC algorithm [14] has been used as guide.

#### 4.3.1.3 Code block segmentation

Code block segmentation is performed as described in [8].

#### 4.3.1.4 Channel coding

The channel coding implemented, is of the type convolutional coding and can handle the rates 1/3 and 1/2 with respective polynomials  $G0 = 561$ ,  $G1 = 753$  and  $G0 = 557$ ,  $G1 = 663$  and  $G2 = 711$  as described in [8]. It works on a bit basis and has lookup tables to get the right 3 or 2 bit values for every shift register state (9 bits). The tables needs  $512*2 + 512*3 = 2560$  bytes of memory, but it speeds thing up by a factor of 9.

[15] has been used as a reference when creating the convolutional encoder.

#### 4.3.1.5 Radio frame equalization

The Radio frame equalization makes sure that the output can be segmented into  $F_i$  equally sized blocks. It does this by padding the input with 0s or 1s.

#### 4.3.1.6 1<sup>st</sup> interleaving

First interleaving is optimized from the functionality described in [8], only two loops are needed and all bit positions are calculated individually. It takes the input and interleaves it with the permutation pattern described in [8].

#### 4.3.1.7 Radio frame segmentation

Radio frame segmentation splits the input sequence into  $F_i$  equal sized segments.

#### 4.3.1.8 Rate matching

The rate matching algorithm is implemented as described in [8]. It takes a frame from every active TrCH, that is, one of the  $F_i$  segments created by the radio frame segmentation. Puncture or repeat these to fit in a radio frame. The SF and number of PhCHs needed is also determined here.

### 4.3.2 TrCH Multiplexing

The TrCH multiplexing takes the rate matched frames and serially multiplexed into one frame called CCTrCH.

### 4.3.3 DPDCH

#### 4.3.3.1 2<sup>nd</sup> interleaving

Just like the 1<sup>st</sup> interleaving, second interleaving is an optimization from the functionality described in [8]. Only two loops are needed and all bit positions are calculated individually, as are the pruned bits. It takes the input and interleaves it with the permutation pattern also described in [8].

#### 4.3.3.2 Physical channel mapping

In this function, the bits in every PhCH are mapped so that they will be transmitted over the air in ascending order.

### 4.3.4 DPCCH

The structure of DPCCH is built up as described in [7]. The sizes of the fields are determined by the slot format received from higher layers in the test bench. The pilot

bits are taken from a table in [7], and the TFCI bits for all slots in one radio frame are calculated using the (32, 10) sub-code of the second Reed-Muller code in [8]. The bits for all other fields are randomly generated for every slot. The DPCCH function returns one slot of DPCCH data.

### 4.3.5 DPCH

Both the DPDCH and the DPCCH are modulation mapped, spread with a channelization code, multiplexed and scrambled.

#### 4.3.5.1 Modulation mapping

This function converts the bits to real values. This is done in a loop, and the way that took fewest clock cycles was to have an if statement that checked if it was a 1 or a 0 and converted it to the corresponding real value.

#### 4.3.5.2 Channelization

The channelization uses lookup tables for the spreading codes which are precalculated using the recursive algorithm in [9]. This reduces Central Processing Unit (CPU) usage and can be done since only one code for every SF is needed when the number of DPDCHs are limited to 1 per user, which is the case for this implementation. The function takes the input and multiplies every bit in it with a spreading code using two nested loops. The length of the output will be SF times longer than the input.

#### 4.3.5.3 Channel multiplexing and I/Q mapping

DPDCH 1 is placed on the I branch. The DPCCH is placed on the Q branch.

#### 4.3.5.4 Scrambling

The scrambling is done by complex multiplication with a scrambling code, Equation 4.1.

$$(a + bi)(c + di) = (ac - bd) + j(bc + ad) \quad (4.1)$$

Scrambling shall be done by a FPGA and VHDL code for this is implemented and simulated in Modelsim.

## 4.4 Confines

Due to the complex structure of the air interface this implementation has been confined so that the master's thesis could be completed in time. Implementation of the system-ized test environment cover baseband encoding with support of DCH users and only has

support for signaling and conversational speech services. No turbo coder is implemented therefore interactive PS and streaming data services can not be used. The implementation does not support compressed mode and has a maximum of 1 DPDCH per user. Some simplifications and hard coding are done in the encoding functionality, but the implementation is designed so that it can easily be extended and made more dynamic.

## 4.5 Verification

The encoder has been verified in different ways. Some parts, like the convolutional encoder and the interleaver are verified with MATLAB. The CRC has been verified using a reference CRC from [16]. Most bit manipulations have been verified by printing the result and compare it with what it should be. The code for most of the functions have been compared with the reference model.

More verification will be done using the reference model, when this has been modified so it can generate data dumps at desired places. When the real-time test environment is completed the complete system will also be verified using the reference model.



---

# CHAPTER 5

---

## Result

### 5.1 Baseband encoder

A new real-time test environment has been systemized and the baseband encoding block has been implemented in software.

The baseband encoder is implemented using a DSP and a FPGA. The DSP application is implemented in C using Code Composer Studio (CCS). VHDL code for the FPGA is simulated in Modelsim. Division of functionality between these two is decided so that the DSP does all encoding up to the scrambling, which will be done by the FPGA.

The application supports multiple DCH users with a maximum one PhCH for each user. It has full support for all Conversational/speech and Signaling Radio Bearer (SRB) RABs. The design is implemented in such a way that it is simple to add functionality to the application without having to edit existing software. The application has been stability tested on existing hardware. EMIF and QDMA transfers are also tested.

Verification of the system has been done in the matter possible. By using MATLAB, existing C code and a reference model, most of the functionality was tested.

### 5.2 Analysis

A speed analysis has been performed on the baseband encoder using CCS cycle-accurate simulator. The CPU cycles are measured for one user, encoding one TTI of data, creating one CCTrCH frame and one slot of PhCH data.

In Table 5.1 a Conversational/speech and SRB RAB has been used. As can be seen it takes about 2.3 % of the total CPU usage<sup>1</sup> to encode it.

---

<sup>1</sup>Total CPU usage is based on how much information that can be sent under 10 ms (one radio frame).

Table 5.2 shows analysis for two RABs including interactive PS services. These RABs requires turbo coding, but since this feature is not implemented yet it is replaces with convolutional coding with rate 1/3. These tests were not actually used for speed analysis, more to see that the functionality in the encoder could handle more and bigger TBSs.

RAB	Cycles	Time [ms] <sup>a</sup>	CPU usage [%]
Conversational/speech 12.2 kbps CS RB + 3.4 kbps SRB	194450	0.23	2.3

*Table 5.1: Simulated encoding of Conversational/speech and SRB RAB.*

---

<sup>a</sup>Assuming a clockrate of 850 MHz

RAB	Cycles	Time [ms] <sup>a</sup>	CPU usage [%]
Conversational/speech 12.2 kbps CS RB + Interactive 64 kbps PS RB + 3.4 kbps SRB	575140	0.68	6.8
Conversational/speech 12.2 kbps CS RB + Interactive 384 kbps PS RB + 3.4 kbps SRB	895001	1.05	10.5

*Table 5.2: Simulated encoding of Conversational/speech, Interactive PS and SRB RABs.*

---

<sup>a</sup>Assuming a clockrate of 850 MHz

## 5.3 Conclusion

A part of the complete system has been implemented, baseband encoding. It can generate I/Q data in real-time and has support for multiple DCH users. Encoding radio frames for a number of users can be done under 10 ms and functionality is divided between a DSP and a FPGA.

---

# CHAPTER 6

---

## Discussion

### 6.1 Design choices

The application is constructed of three main modules, which all have their own coding module. The modules represents the three channels available, TrCH, CCTrCH and PhCH. All functions are independent of each other and can be used freely in the design. This assembly is chosen because it gives a good overview of the system, it is easy to edit existing functionality, easy to add new functionality and gives good debugging properties.

The users in the test environment have three state variables, just like the main modules they have one for every channel. Each state variable includes all parameters needed for encoding of that channel. The state variables also includes buffers for encoded data. This way everything can be encoded in advance, which is required to hold the timing requirements.

Most of the functionality is placed on the DSP. It has a good development kit and with the use of a Blackhawk LAN560 JTAG Emulator it is very easy to do real-time debugging. By using the highest optimization and the right compiler keywords a very fast and accurate encoding can be achieved.

The FPGA does the scrambling and generation of the scrambling code. This decision is based on several test, where it has been realized that when more users enters the system the DSP memory will not be enough for efficient encoding. The scrambling is the last step in the baseband encoding chain and therefore no data has to be returned to the DSP, so no transfer time is lost there.

To confine the implementation for this master thesis is necessary to be able to finish it in time. The decision for implementing convolutional coding is based on the fact that all DCH RABs includes a SRB TrCH which is convolutional coded and that it can easily be verified in MATLAB. Another limit of the implementation is that there can only be

one PhCH per user. This design choice simplifies some of the algorithms in the encoding chain and is not limiting the baseband encoder.

## 6.2 Future work

The baseband encoder part on the DSP will be integrated in OSEck. To complete the real-time test environment, TX and RX analog parts have to be designed. These parts are to be implemented on the FPGA. The test environment will be integrated in the test bench and an interface outwards will be designed. When all parts are in place the test environment shall be tested and verified.

To support RABs that includes PS TrCHs the turbo coder has to be implemented and enable support for more than DCHs and allow more than one PhCH per user.

More work on speed optimization can be done, where the division of functionality between the DSP and FPGA is an important and complex part. Much time must be spent on testing if it is worth sending data to the FPGA and back for some functionality, or if it is better to do it on the DSP.

---

# APPENDIX A

---

## Error Coding

### 1.0.1 CRC

Cyclic Redundancy Check (CRC) is a block code and manipulates data one block at a time. The CRC encoder generates redundancy bits that are added to the end of each data block. With these redundancy bits the receiver can check if errors has occurred during transmission.

In a block coder the output bits  $n$  is always larger than the input bits  $k$ . This is refereed to as an  $(n, k)$  code, where the code rate is  $n/k$  and the length of the redundancy bits are  $n-k$ . The redundancy is calculated using Equation A.1. The longer this sequence is the more errors it can detect.

$$\frac{n - k}{n} \tag{A.1}$$

There exists  $2^k$  possible information blocks that can be mapped onto  $2^n$  code words. Since  $n$  is bigger than  $k$  most of the code words will not be used. The ones that are used are chosen to maximize the performance of the channel decoder.

The CRC is calculated by dividing the input with a given polynomial using modulo-2 division. The remainder will then be the CRC checksum. Available polynomials for Universal Mobile Telecommunications System (UTMS) can be of degree 24, 16, 12, 8 and 0, Table A.1. The start reminder is 0 and the result reminder is reflected.

Nr of CRC bits	Polynomial
24	$D^{24} + D^{23} + D^6 + D^5 + D + 1$
16	$D^{16} + D^{12} + D^5 + 1$
12	$D^{12} + D^{11} + D^3 + D^2 + D + 1$
8	$D^8 + D^7 + D^4 + D^2 + D + 1$

Table A.1: CRC polynomials for UMTS

For example,  $D^8 + D^7 + D^4 + D^2 + D + 1$  will give a divisor of 110010111.

## 1.0.2 FEC

### 1.0.2.1 Convolutional codes

Convolutional codes are error correcting codes that will both find and correct errors. These codes are often used when retransmission is not suitable. The convolutional coders used in UMTS can be seen in Figure A.1.

An convolutional coder includes the input data in the output. This greatly improves Bit Error Ratio (BER) performance, at the cost of more data to transmit. The encoding algorithm takes  $m$  input symbols and converts them to  $n$  output symbols.  $m/n$  is called the code rate where  $n \leq m$ . The transformation is a function of the last  $k$  bits, where  $k$  is the constraint length of the code. The notation of the encoder is said to be  $(n, m, k)$ . In UMTS these numbers are  $(3, 1, 9)$  for rate  $1/3$  and  $(2, 1, 9)$  for rate  $1/2$ .

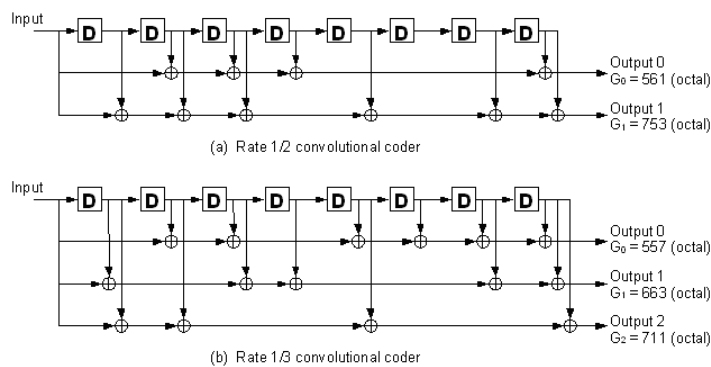


Figure A.1: Rate  $1/2$  and rate  $1/3$  convolutional coders

The encoder can be seen as a series of  $m-1$  shift registers. The shift registers are tied

to modulo-2 adders according to the 1s in a generator polynomial  $G^1$ . The initial state of the encoder has to be all zeroes. To ensure this the shift registers are filled with  $m-1$  zeroes after each encoding.

The correcting capability  $t$  of a convolutional code is the number of errors that can be corrected by the code.  $t$  can be calculated using Equation A.2.

$$t = \left\lfloor \frac{d-1}{2} \right\rfloor \quad (\text{A.2})$$

where  $d$  is the free distance, a minimal Hamming distance<sup>2</sup> between different encoded sequences.

### 1.0.2.2 Turbo codes

Turbo codes are high-performance error correcting codes developed in 1993. The big advantage of the turbo codes is that they come very close to approaching the Shannon limit<sup>3</sup>.

This makes it possible to increase the data rate without increasing the power of the transmission, or decrease the power but use the same data rate.

Turbo codes have best efficiency with high data rate services. When low data rates are used the decoding process is power-intensive in respect to the number of bits received. This is also the reason why Turbo codes are unusable in long-term communication. The ideal usage for Turbo codes are in Packet Switched (PS) data channels, which are only used for short periods of time, but at a very high peak data rate.

The main functionality of the turbo coder is that it uses the output data to readjust the input data. This iterative process will improve the quality of the decoder output.

The turbo coder used in the UMTS can be seen in Figure A.2. It is a Parallel Concatenated Convolutional Code (PCCC) and consists of two Recursive Systematic Convolutional (RSC) coders in parallel separated by an interleaver. The task of the interleaver is to randomize the data before it enters the second encoder.

<sup>1</sup>Note that these polynomials are given in octal form.

<sup>2</sup>The number of positions for which the symbols in two strings of equal size are different.

<sup>3</sup>Theoretical limit of maximum information transfer rate over a noisy channel.

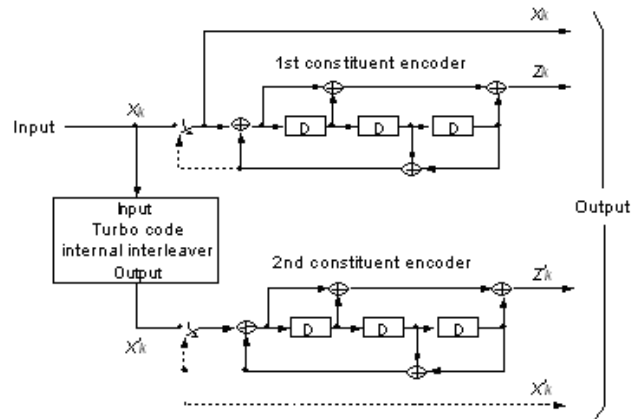


Figure A.2: Rate 1/3 turbo coder

All algorithms for UMTS Forward Error Correction (FEC) coding can be found in [8].

### 1.0.3 Interleaving

Interleaving is a way to arrange the data in a non-contiguous way to make it stand burst errors. These types of errors can destroy many bits in a row and make it hard to recover using FEC coding, since these expects the errors to be more uniformly distributed.

This method is popular because it is a less complex and cheaper way to handle burst errors than directly increasing the power of the error correction scheme. An example of a block interleaver can be seen in Figure A.3.

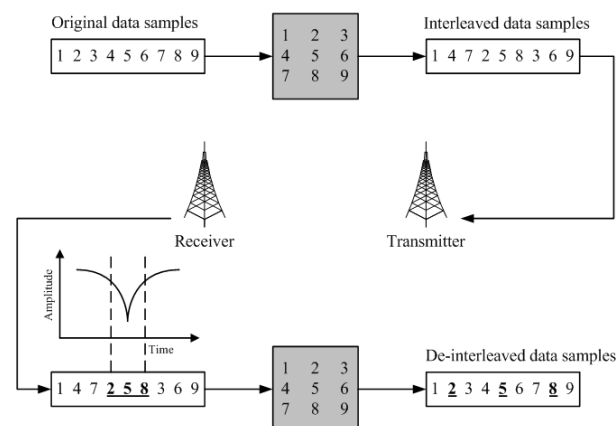


Figure A.3: Block interleaver

---

# **APPENDIX B**

---

## **Protocols**

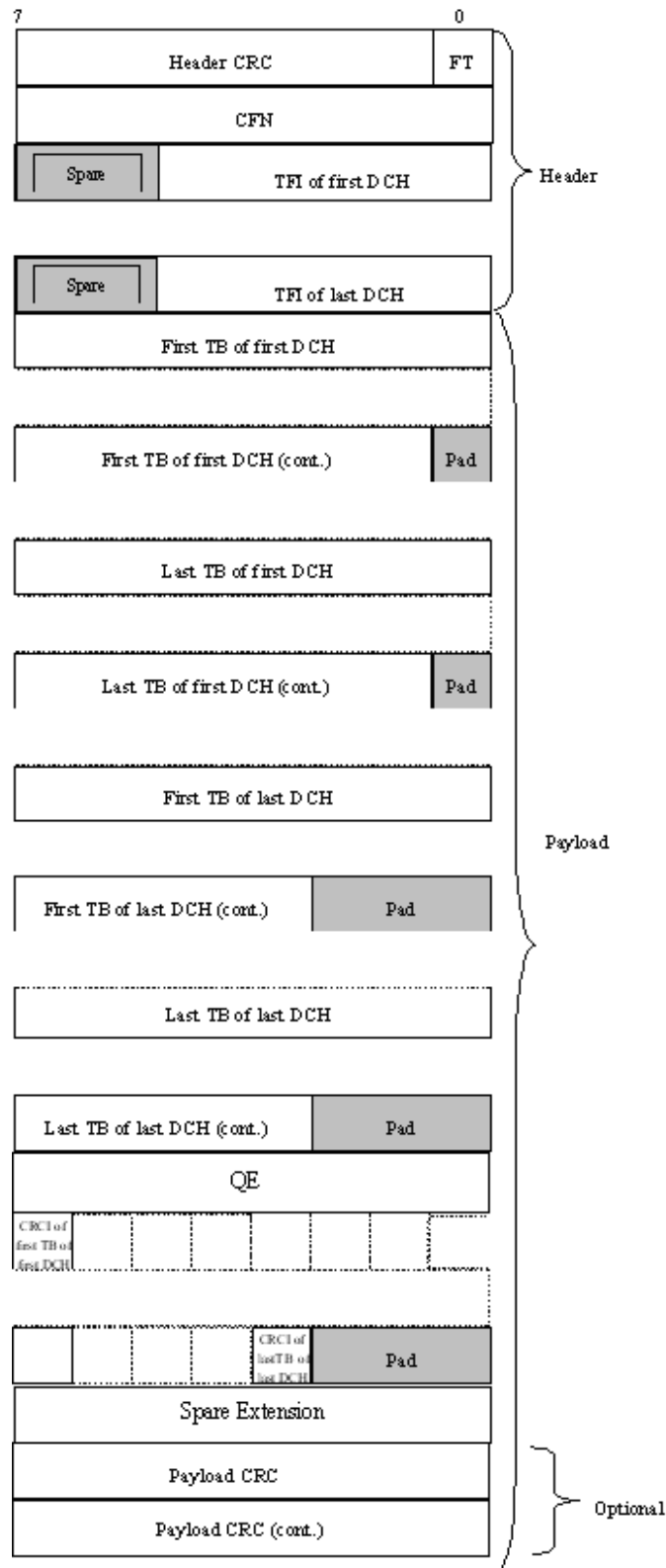


Figure B.1: Uplink data frame structure

---

# **APPENDIX C**

---

## **Charts**

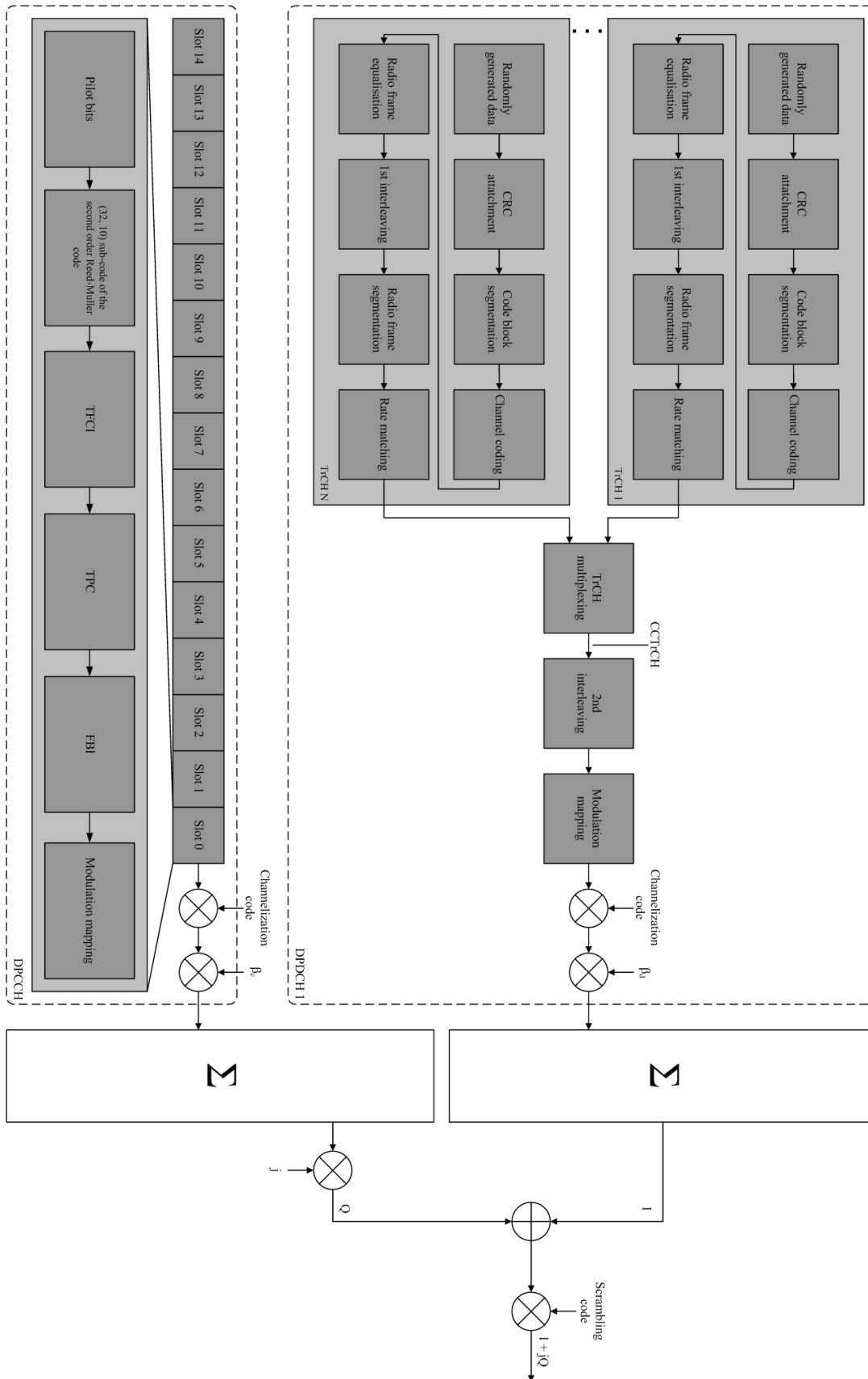


Figure C.1: Baseband encoder flow chart

---

## REFERENCES

---

- [1] Wikipedia, “Cellular network — wikipedia, the free encyclopedia.” [http://en.wikipedia.org/w/index.php?title=Cellular\\_network](http://en.wikipedia.org/w/index.php?title=Cellular_network), 2007. [Online; accessed 23-October-2007].
- [2] 3GPP, “Utran overall description (release 6),” Tech. Rep. 25.401, 3rd Generation Partnership Project, 2005.
- [3] H. Holma and A. Toskala, *WCDMA for UMTS*. Wiley, 2001.
- [4] 3GPP, “Physical layer - measurements (fdd) (release 6),” Tech. Rep. 25.215, 3rd Generation Partnership Project, 2005.
- [5] 3GPP, “Synchronisation in utran stage 2 (release 6),” Tech. Rep. 25.402, 3rd Generation Partnership Project, 2005.
- [6] 3GPP, “Services provided by the physical layer (release 6),” Tech. Rep. 25.302, 3rd Generation Partnership Project, 2005.
- [7] 3GPP, “Physical channels and mapping of transport channels onto physical channels (fdd) (release 6),” Tech. Rep. 25.211, 3rd Generation Partnership Project, 2005.
- [8] 3GPP, “Multiplexing and channel coding (fdd) (release 6),” Tech. Rep. 25.212, 3rd Generation Partnership Project, 2005.
- [9] 3GPP, “Spreading and modulation (fdd) (release 6),” Tech. Rep. 25.213, 3rd Generation Partnership Project, 2005.
- [10] T. Instruments, *TMS320C6414T, TMS320C6415T, TMS320C6416T FIXED-POINT DIGITAL SIGNAL PROCESSORS*, 2008.
- [11] T. Instruments, *External Memory Interface (EMIF) Reference Guide*, 2006.
- [12] T. Instruments, *Enhanced Direct Memory Access (EDMA) Controller Reference Guide*, 2005.

- [13] Xilinx, *Virtex-5 FPGA User Guide*, 2008.
- [14] M. Barr, “Crc implementation code in c.” <http://www.netrino.com/Embedded-Systems/How-To/CRC-Calculatation-C-Code>, 2000.
- [15] C. Fleming, “A tutorial on convolutional coding with viterbi decoding.” <http://home.netcom.com/~chip.f/viterbi/tutorial.html>, 2006.
- [16] M. Isaksson and U. Andersson, “Downlink baseband decoder implementation,” Master’s thesis, 2007.