Specification

FlexLink Physical Layer Definitions

|  |  |
| --- | --- |
| Version: | 00.08 |
| Date: | Nov 28, 2022 |
| Author: | Andreas Schwarzinger |
| Status: | Preliminary |

Revision History

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Contents** |
| 00.01 | Aug 20, 2022 | Basic Overview |
| 00.02 | Aug 31, 2022 | Basic Overview |
| 00.03 | Sept 15, 2022 | Basic Overview |
| 00.04 | Sept 23, 2022 | Added *PreambleA* Definition |
| 00.06 | Oct 8, 2022 | Added MAC Definitions |
| 00.07 |  |  |
| 00.08 | Nov 28, 2022 | Added *PreambleB* Definition |
| 00.09 |  | Added a register map |
| 00.10 | Apr 23, 2023 | remove data link layers and MAC chapters |

Table of Contents

1. FlexLink Overview 3

1.1 Introduction 3

1.1.1 Definitions of Terms in this Document 3

1.1.2 Method of Modulation 4

1.1.3 Subcarrier Spacing 4

1.1.4 Cyclic Prefix Length 4

1.1.5 Reference Clocks 4

1.1.6 Bandwidth 5

1.1.7 Modulation and Coding Scheme 5

1.1.8 Antenna Configuration 6

2. Packet Construction 7

2.1 Basic Packet Layout 7

2.1.1 Preamble 7

2.1.2 OFDM Symbol Construction 8

2.2 Resource Grid Construction 8

2.2.1 Demodulation Reference Symbol Rate 9

2.2.2 Demodulation Reference Symbol Periodicity 10

2.2.3 Demodulation Reference Symbols 10

2.2.4 Phase Noise Reference Signals 11

2.2.5 The Control Information 12

2.2.6 Data Symbols 12

2.2.7 The DC Gap 12

2.2.8 QAM Symbol Mapping Definitions 13

2.2.9 Interleaving 15

2.3 The Signal Field 16

2.4 The Payload Fields 16

3. Preamble Definitions 18

3.1 The AGC Burst 18

3.2 PreambleA 19

3.2.1 Packet Detection 20

3.2.2 Frequency Offset Detection 21

3.3 PreambleB 24

4. Transmitter Architecture 27

4.1 Transmit Diversity 27

5. Data Link Layer (DLL) 28

6. Transaction Layer (TL) 31

7. OFDM comparisons 32

7.1 Comparison table 32

8. Appendix 34

# FlexLink Overview

## Introduction

FlexLink is a physical layer and MAC layer specification for a packet based TDD (time division duplex) radio link that supports both high reliability and high throughput transmission in scenarios where the transmit and receive terminal may be in motion. The physical layer will borrow concepts from both 802.11 WLAN and 4G/5G cellular technologies in order to provide a multipurpose point-to-point and point-to-multipoint link that can be used in a variety of wireless communication applications.

This specification will be organized base on the following premise.

🡪 A basic set of features that must be developed first and be part of all compliant modem implementations.

🡪 An expanded set of features that should be implemented in addition to the compliant capabilities to provide performance improvements and an expanded application range.

### Definitions of Terms in this Document

|  |  |
| --- | --- |
| **Term** | **Explanation** |
| Forward Link | Transmission direction from the control station to the mobile station. |
| Reverse Link | Transmission direction from the mobile station to the control station. |
| OFDM | Orthogonal frequency division multiplexing |
| Resource Element | A single OFDM tone (subcarrier) orientation in a single OFDM symbol. |
| Demodulation Reference Symbol | An OFDM symbol containing reference or pilot information used for the purpose of channel estimation and subsequent equalizer programming |
| Phase Reference Signals | The reference (pilot) signals are located in every data OFDM symbols and help reject phase noise, which is a problem at high RF frequencies. |
| Modulation and Coding Scheme | In many specifications, the MCS (modulation and coding scheme) represents and index that refers to a configuration of FEC encoding, rate matching and QAM constellation. The lower the MCS index the more robust the link. In this specification, the MCS is simply a term referring to the combination of FEC encoding, rate matching and QAM constellation but does not assign any index or restrict the combination. |
| Transport Block | A transport block is defined as a block of bits that can be conveniently encoded by the LDPC encoders available in the specification. A transport block will consist of data bits from the MAC and a 24-bit CRC protecting these data bits. The CRC allows the MAC to verify that the data bits have arrived intact. The legal transport block sizes are described later in this document. |
|  |  |

### Method of Modulation

The method of modulation in both the forward and reverse direction shall be orthogonal frequency multiplexing, or OFDM.

### Subcarrier Spacing

The subcarrier spacing should be chosen based on the following criteria.

1. Significant multipath requires a smaller subcarrier spacing in order to keep the frequency response of the channel constant across each of them.
2. Poor crystal oscillator specifications (large frequency offsets) and poor phase noise performance at high center frequencies require a larger subcarrier spacing.

*Mandatory:* 20KHz (The preliminary specification will focus on the 20KHz case)

*Optional:* 40KHz, 80KHz and 160KHz

### Cyclic Prefix Length

The cyclic prefix length avoids inter-symbol interference and depends on the difference in delay time between the paths in our channel. LTE had initially chosen a normal CP of about 5 microseconds and an extended CP of 16 microseconds. Given that the extended CP was never used, the mandatory cyclic prefix is chosen to be 4 microseconds.

*Mandatory:* 4 microseconds (The preliminary specification will focus on the 4 microseconds case)

*Optional:* 1, 2, 8 microseconds

For RF links operating at low RF frequencies in rural and hilly environments, 8 microseconds are warranted. For indoor applications, 1 microsecond is warranted.

### Reference Clocks

*Frequency of Choice*

The frequency of choice would be a master reference clock that is an integer multiple of 20.48MHz yielding a subcarrier spacing of 20KHz when paired up with a *N*=1024 sized IFFT for OFDM modulation. Down sampling and any associated filtering can be done at the higher sampling frequencies, whereas the 20.48MHz is the base rate at which the I/FFTs operate.

🡪 Subcarrier Spacing = 20.48e6/1024 = 20.0KHz

In addition to these, alternate master reference clocks could be integer multiples of 20.00MHz and 19.20MHz, yielding the following subcarrier spacing if the IFFT size is 1024. (The preliminary version of this specification assumes a master clock that is an integer multiple of 20.48MHz)

🡪 Subcarrier Spacing = 20e6/1024 = 19.53125KHz

🡪 Subcarrier Spacing = 19.2e6/1024 = 18.75KHz

*Ideal Frequency Stability*

The ideal frequency stability would be one where the frequency offset between the transmit LO and receive LO is no more than 3KHz, which at a 6GHz center frequency represents a combined (TX/RX) stability of approximate 0.5ppm. The offset must be such that a zero-IF analog receiver, which will destroy information at DC, will not affect subcarriers at *Tone* 1 and *Tone* -1.

*Realistic Frequency Stability*

If master clocks with stabilities under 0.5ppm are not available or not realistic in terms of cost, than less stable oscillators may be used with the caveat, that a frequency synchronization packet must first be transmitted allowing the offset to be detected and the RX synthesizer to be reprogrammed to match the TX LO properly before proper demodulation.

### Bandwidth

Four different bandwidths are proposed, including the mandatory 20MHz channel, as well as the option 5MHz, 10MHz, and 40MHz channels.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | SC = 20KHz, IFFT = 50usec, CP = 4usec | | | | | |
|  |  | **5** | **10** | **20** | **40** |
| **FFT Size** |  |  | 256 | 512 | 1024 | 2048 |
| **Fs (MHz) for FFT** |  |  | 5.12 | 10.24 | 20.48 | 40.96 |
| **ADC Sample Rate (MHz)** |  |  | Fs=n‧10.24 | Fs=n‧20.48 | Fs=n‧40.96 | Fs=n‧81.92 |
| **CP Length Samples** |  |  | 20 | 41 | **82** | **164** |
| **Number of Subcarriers** |  |  | 219 | 442 | 901 | 1801 |
| **BW (MHz) at RF** |  |  | 4.4 | 8.8 | 18 | 36 |
| **BW (MHz) at BB** |  |  | 2.2 | 4.4 | 9.0 | 18 |

Figure ‑: FlexLink Configuration for Subcarrier Spacing of 20KHz and CP = 4 microseconds

### Modulation and Coding Scheme

The modulation and coding scheme, or MCS, determines the QAM constellation that should be used as well as the amount of FEC and the rate matching, which in FlexLink reduces to simple repetition of bits. MCS simply refers to a group of three configurations (QAM constellation, FEC configuration, and rate matching index).

#### QAM Constellations

The Flex link should support BPSK, QPSK, 16-QAM and 64-QAM with the same bit mapping proposed in the IEEE 802.11A standard.

#### FEC – Forward Error Correction

The signal field will use a 256-bit polar encoder / decoder coding. The payload should support the LDPC rate ½, 2/3, and ¾ encoders / decoders that are provided in the WLAN 802.11n specification.

#### Rate Matching

Rate matching will repeat the array of transport bits that have been FEC encoded. We use a rate matching index featuring 2 bits to indicate how to execute rate matching. Note, the number of bits in any transport block should always be an integer multiple of 8 bits.

|  |  |
| --- | --- |
| **RI** | Explanation |
| **0** | No rate matching. |
| **1** | Each bit shall be repeated once. |
| **2** | Each bit shall be repeated twice. |
| **3** | Each bit shall be repeated four times. |

Table ‑: Available Rate Matching

### Antenna Configuration

The following antenna configuration should be considered.

1. (Mandatory) The SISO (single input single output) configuration featuring a single transmit and a single receive antenna. This is the easiest and lowest performance antenna configuration.
2. (Optional) The MISO (multiple input single output) is used to improve the channel and provides better signal to noise ratio. It is not terribly difficult to implement. LTE uses this all the time.
3. (Optional) The SIMO (single input multiple output) is simply a maximum ratio type combiner improving the channel and signal to noise ratio just like the MISO case.
4. (Option) The 2x2 MIMO configuration increases the throughput of the system by a maximum of two-fold. Alternatively, MISO and SIMO can be combined for the most robust channel and signal to noise ratio.

# Packet Construction

## Basic Packet Layout

The FlexLink is a TDD type link that will transmit packets in both the forward and reverse direction.



Figure ‑: Basic Packet Structure for FlexLink

The FlexLink specification provides a preamble, a signal field as well as two payload fields: Payload *A* and *B*. The payloads are separated such that each can have a different modulation and coding schemes and the packets can thus convey two messages with different reliability. For example, payload *A* can feature control information that is more important for the receiver than data information in payload *B*.

### Preamble

The preamble is composed of three separate portions, which help to acquire the following quantities. The different portions of the preamble are defined in chapter three.

***AGC Burst***

🡪 The initial portion of the preamble is a four microsecond long wideband AGC (automatic gain control) burst that facilitates the convergence of all gain stages in the analog radio portion of the modem with the goal of providing IQ data that is within the proper ADC input range.

***Preamble A***

Preamble A shall consist of a 512-sample waveform if one packet detection is desired, or a 4096 + 256 sample waveform if both packet detection and frequency offset correction is desired.

🡪 Packet detection – This algorithm discerns a FlexLink packet from any other that might legally be transmitted in the band.

🡪 Frequency offset acquisition – The frequency offset must be acquired and corrected in order to avoid loss of orthogonality in the OFDM demodulator.

***Preamble B***

🡪 FFT Timing determination – Preamble *B* will consist of a sequence that will have excellent autocorrelation properties with distinct peaks that allows the receiver to determine the arrival time of the strongest RF path. The Preamble *B* is one OFDM symbol in length.

### OFDM Symbol Construction

Information that is to be OFDM modulated is organized into a resource grid and then mapped into a length *N* = 1024 IFFT. The organization of that data is defined in section 2.2. Of the 901 information bearing subcarriers, 451 are mapped into the positive frequency inputs at IFFT[0] through IFFT[450], whereas the remaining 450 are mapped into the negative frequency inputs at IFFT[574] through IFFT[1023]. The last 82 samples of the IFFT output sequence shall be copied and prepended as the cyclic prefix. For a sample rate of 1/20.48MHz, the cyclic prefix features a default length of 82/20.48MHz = 4.004 microseconds.



Figure ‑: OFDM Modulation using a Length 1024 IFFT

## Resource Grid Construction

The resource grid, which is a 3GPP construct used in 4G LTE and 5G New Radio, is a nice illustration of the distribution of reference signals, control information and data. In the 4G/5G world, the subcarriers in the resource grid are marked with the variable *k*, where *k* = 0 represents the most negative subcarrier. In the IEEE world, the term tone is used, which may be negative or positive. Both conventions are shown in the resource grid diagram. The DC carrier is part of the resource grid. Note that the reference signals are spaced with a period of three subcarriers in the frequency domain whereas the spacing in the time domain is programmable. They are spaced evenly such that the Fourier transform may be employed during channel estimation. The resource grid contains resource elements, where one resource element references a single subcarrier of a single OFDM symbol. The following types of information is placed into these resource elements.

🡪 Reference signals (pilot information) for transmit antenna port 0.

🡪 Reference signals (pilot information) for transmit antenna port 1.

🡪 Control information providing configuration information regarding the resource grid.

🡪 Reserved resource elements that are to be set to a value of 0 + j0.



Figure ‑: Reference Grid for the 20MHz Bandwidth Case and Subcarrier Spacing of 20KHz

### Demodulation Reference Symbol Rate

Reference symbols are OFDM symbols that contain reference signal information. This reference signal information is required to estimate the multipath channel and program the equalizer. The spacing of the reference symbols depends on the amount of unresolved frequency offset and Doppler shift. Assuming that no frequency offset exists due to differences in LO frequencies, the maximum frequency offset due to Doppler will be related to a highest supported speed differences between the transmit and receive terminals. The highest speed difference supported by the specification is 200Kph, or 55 m/sec. The maximum Doppler frequency is computed as follows for a center frequency of 5.9GHz.



Table ‑: Repetition of Reference Signal OFDM Symbols

The reference signal symbols should therefore appear every 0.25 / 983 Hz, or approximately every 250 microseconds. However, as the Doppler appears as a frequency offset, it will be acquired during preamble processing and higher Doppler shifts can easily be processed, and symbols spacing can often be chosen to be more generous. The reference symbol spacing will be programmable.

### Demodulation Reference Symbol Periodicity

As mentioned in section 2.2.1, demodulation reference symbols appear at a rate that allows us to know the channel in the presence of unresolved frequency offset and Doppler. The periodicity is selectable via the control information. A value of 0 indicates that only the first demodulation reference symbol shall be transmitted usually followed by phase noise reference signals.

|  |  |
| --- | --- |
| **Subcarrier Spacing** | **Available Demodulation Reference Symbol Periodicity** |
| **20KHz** | 1 ‧ [0, 2, 4, 6, …, 30] |
| **40KHz** | 2 ‧ [0, 2, 4, 6, …, 30] |
| **80KHz** | 4 ‧ [0, 2, 4, 6, …, 30] |
| **160KHz** | 8 ‧ [0, 2, 4, 6, …, 30] |

Table ‑: Periodicity of Demodulation Reference Symbols for Different Subcarrier Spacing

### Demodulation Reference Symbols

The demodulation reference OFDM symbols will contains reference signals (pilots) that are known ahead of time to the receiver. The demodulation reference symbols will contain resource elements that are used for reference signals, control information as well as data. The spacing between reference signal shall be three resource elements. The tighter the spacing, the easier it will be to reject noise from the channel estimate.

#### The First Demodulation Reference Symbol and Layout of Control Information

🡪 This specification allows for one (port 0) or two (port 0 and port 1) transmit antenna ports, all of which must send demodulation reference signals. The resource elements into which TX Port 0 maps its demodulation reference signals must be avoided (left empty) by TX Port 1, and visa-versa. This way, a single receiver can estimate the channel for each port independently. In the case that only one TX antenna exists (Port 0), its reference signal magnitude shall be boosted by sqrt(2).

🡪 The 12 control information bits are **only** sent from Port 0 and are BPSK mapped into the resource elements assigned to them. Note that there is one resource element available for control information for each triplet (block of 3 resource elements within a single OFDM symbol). To use all resource elements available for control information properly, we will map each control information bit into a row vector of length *N* as follows.

0 🡪 [0, 0, 0, 0, 0, 0, 0, 0, …… 0]

1 🡪 [1, 1, 1, 1, 1, 1, 1, 1, …… 1]

The length of each mapped vector should be computed as follows.

Thus, if the bandwidth has 300 triplets, and we wish to map 12 control bits, then the vectors will feature a length of 25. The 12 vectors with 25 bits each must now be flattened into a single vector by concatenating each column. Assume the 12 vectors are arranged along the rows of a matrix as follows.

The new flattened vector will look as follows.

Each element in the vector *V* shall be mapped consecutively into the resource elements available for control information starting from the most negative tones to the most positive tones.

#### Remaining Demodulation Reference Symbols

The remaining demodulation reference symbols do not carry control information. They carry data QAM symbols in all resource elements that do not carry demodulation reference signals from the TX antenna ports. Clearly, for a single TX antenna port, more resource elements may be dedicated to data.

### Phase Noise Reference Signals

Phase noise reference signals are provided for situations in which neither terminal is in motion. In this case, the frequency response is altered by unresolved frequency offset, phase noise and timing drift only. Aside from the initial demodulation reference symbol, further reference symbols may not be required to determine the changing channel conditions. Phase noise reference signal may be used instead. Phase noise reference signals are place according to the following rules.

🡪 Phase noise reference signals for TX port 0 shall be placed at the following eight tones: [±4, ±3, ±2, ±1] ‧ 90 = ±360, ±270, ±180, ±90.

🡪 Phase noise reference signals for TX port 1 shall be placed at the following eight tones: [±4, ±3, ±2, ±1] ‧ 90 -1 = [-361, -271, -181, -91, 89, 179, 269, 359]

🡪 Phase noise reference signals shall overwrite data resource elements in OFDM symbols that are not demodulation reference symbols.

🡪 All phase noise reference signals shall feature the BSPK value 1 + 0*j*.

### The Control Information

The control information will be embedded within the first demodulation reference symbol. This information is embedded in this manner so that the transmitting MAC can control some aspects of the link on the fly, without previously having to inform the receiver of the configuration change. This is also important so that the transmitter can communicate in an ad hoc manner with unknown receivers. There are aspects of the link that are not embedded and must be known ahead of time such as the *bandwidth*, *subcarrier* spacing and the *cyclic* *prefix* length. The remaining configuration data is made available via the control information.

|  |  |  |
| --- | --- | --- |
| **Information** | **Number of Bits** | **Description** |
| **Reference Periodicity** | 3 bits | The period in OFDM symbols with which demodulation reference symbols appear in the TX IQ stream.  [0, 1, 2, …, 15] 🡪 [0, 2, 4, 6, … 30] symbols |
| **Number Signal Field Symbols** | 2 bits | The number of OFDM symbols for the signal field.  [0, 1, 2, 3] 🡪 [1, 2, 3, 4] symbols |
| **Number of Tx Antenna Ports** | 1 bit | [0, 1] 🡪 [1, 2] Transmit antennas |
| **Use Phase Reference Signals** | 1 bit | Indicates whether phase reference signals are used in addition to demodulation reference signals. |
| **Use Reference Signal Boosting** | 1 bit | This bit is only meaningful in case of one TX antennas. It indicates whether reference signals on port 0 should be boosted by *sqrt* (2) in magnitude for better signal to noise ratio. If this bit is 1, then data may not be mapped onto the reference elements reserved for reference signals of TX port 1. |
| **BPSK [0] / QPSK [1] in Signal Field** | 1 bit | Use QPSK in the signal field for links with reasonable SNR in order to occupy fewer OFDM symbols. |
| **Reserved** | 3 bits | Bits reserved for future expansion of standard. |
|  | 12 |  |

Table ‑: Understanding Control Information

### Data Symbols

Data symbols will carry QAM data on each subcarrier except for the DC tone and tones reserved for phase reference signals. In general, data symbols are mapped starting at the resource element with the most negative frequency tone to the resource element with the most positive frequency tone.

### The DC Gap

Reference elements at DC (tone 0), shall be set to 0 + *j*0 by the transmitter for each OFDM symbol.

### QAM Symbol Mapping Definitions

Each subcarrier will be scaled by either BPSK, QPSK, 16-QAM or 64-QAM constellation symbols. The mapping process that translates bits to QAM symbols is shown below as a lookup tables followed by a modulation dependent scaling factor.

|  |  |  |
| --- | --- | --- |
| ***Input Bit (b0)*** | ***I Out*** | ***Q Out*** |
| 0 | -1 | 0 |
| 1 | 1 | 0 |

Figure ‑: BPSK Mapping

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Input Bit (b0)*** | ***I Out*** |  | ***Input Bit (b1)*** | ***Q Out*** |
| 0 | -1 |  | 0 | -1 |
| 1 | 1 |  | 1 | 1 |

Figure ‑: QPSK Mapping

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Input Bits (b0 b1)*** | ***I Out*** |  | ***Input Bits (b2 b3)*** | ***Q Out*** |
| 00 | -3 |  | 00 | -3 |
| 01 | -1 |  | 01 | -1 |
| 11 | 1 |  | 11 | 1 |
| 10 | 3 |  | 10 | 3 |

Figure ‑: 16QAM Map

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Input Bits (b0 b1 b2)*** | ***I Out*** |  | ***Input Bits (b3 b4 b5)*** | ***Q Out*** |
| 000 | -7 |  | 000 | -7 |
| 001 | -5 |  | 001 | -5 |
| 011 | -3 |  | 011 | -3 |
| 010 | -1 |  | 010 | -1 |
| 110 | 1 |  | 110 | 1 |
| 111 | 3 |  | 111 | 3 |
| 101 | 5 |  | 101 | 5 |
| 100 | 7 |  | 100 | 7 |

Figure ‑: 64QAM Map

After the mapping process, the complex output values are scaled, depending on the chosen constellation. The BPSK, QPSK, 16QAM, and 64QAM output values are multiplied by the factors 1, 1/*sqrt*(2), 1/*sqrt*(10), and 1/*sqrt*(42), respectively. This is done so that all constellations will have the same *rms* value, thus the same average output power. Although the process is shown as a two-step task, the hardware implementation can combine them into one lookup table per modulation format.

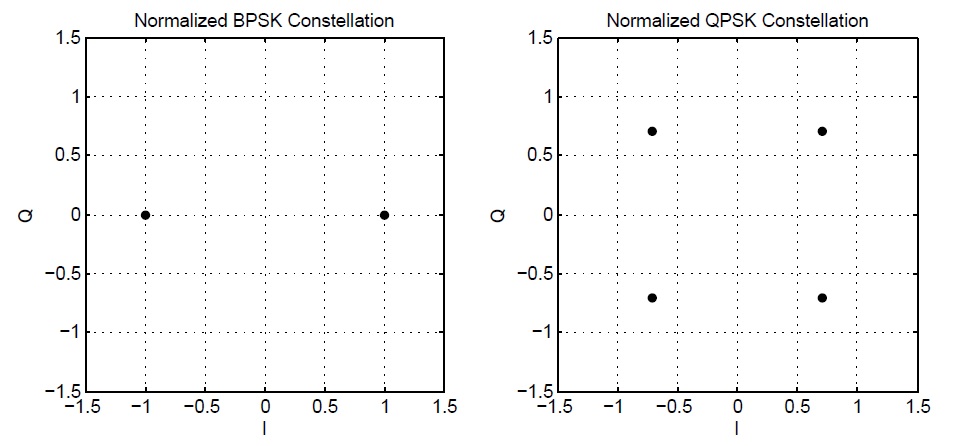


Figure ‑: Normalized BPSK and QPSK Constellations for WLAN 802.11a

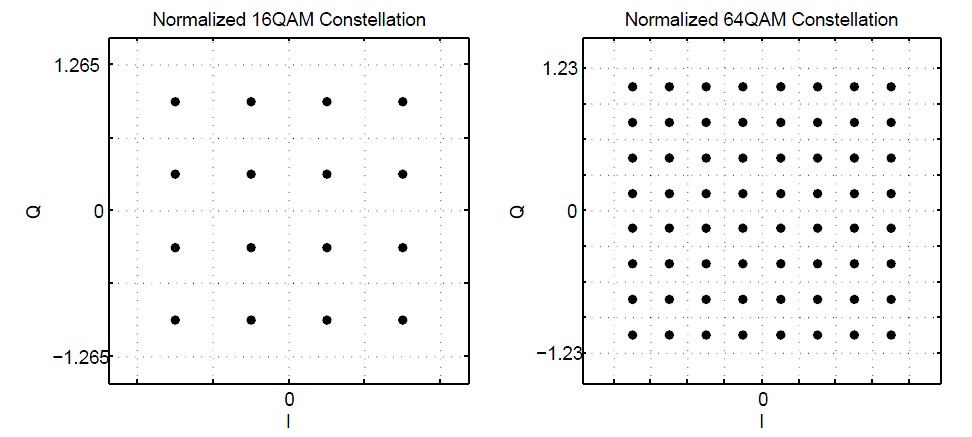


Figure ‑: Normalized 16 and 64 QAM Constellations for WLAN 802.11a

**function** [OutputSymbols] = Mapper\_OFDM(InputBits, BitsPerSymbol)

**% BitsPerSymbol: 1, 2, 4, 6 --> BPSK, QPSK, 16QAM, 64QAM**

**persistent** BPSK\_LUT QPSK\_LUT QAM16\_LUT QAM64\_LUT

**if**(isempty(BPSK\_LUT))

BPSK\_LUT = [-1; 1];

QPSK\_LUT = [-1; 1] / sqrt(2);

QAM16\_LUT = [-3; -1; 3; 1] / sqrt(10); **% Careful Gray Coding**

QAM64\_LUT = [-7; -5; -1; -3; 7; 5; 1; 3] / sqrt(42);

**end**

NumberOfSymbols = floor( length(InputBits)/BitsPerSymbol );

OutputSymbols = zeros(1,NumberOfSymbols);

**for** i = 1:NumberOfSymbols

Start = 1 + (i - 1)\*BitsPerSymbol;

Stop = Start + BitsPerSymbol - 1;

BitGroup = InputBits(1,Start:Stop);

**switch**(BitsPerSymbol)

**case** 1

Symbol = BPSK\_LUT(BitGroup(1,1) + 1, 1);

**case** 2

Symbol = QPSK\_LUT(BitGroup(1,1) + 1,1) + ...

j\*QPSK\_LUT(BitGroup(1,2) + 1, 1);

**case** 4

Symbol = QAM16\_LUT(BitGroup(1,1)\*2 + BitGroup(1,2)+1,1)+ ...

j\*QAM16\_LUT(BitGroup(1,3)\*2 + BitGroup(1,4) + 1,1);

**case** 6

Symbol = QAM64\_LUT(BitGroup(1,1)\*4 + BitGroup(1,2)\*2 + ...

BitGroup(1,3) + 1,1) + ...

j\*QAM64\_LUT(BitGroup(1,4)\*4 + BitGroup(1,5)\*2 + ...

BitGroup(1,6) + 1,1);

**end**

OutputSymbols(1,i) = Symbol;

**end**

### Interleaving

## The Signal Field

The signal field will consist of up to 4 OFDM symbols. The number of OFDM symbols assigned to the signal field is programmable via the control information in the first demodulation reference symbols of the signal field. The signal field will contain vital information for the payload fields that are to follow.

|  |  |  |
| --- | --- | --- |
| **Information** | **Number of Bits** | **Description** |
| **EBS1** | 2 | The encoded block size (648, 1296, 1944 bits) |
| **NTB1** | 16 | The number of transport blocks to be sent in the payload. A value of 0 is not valid. |
| **FEC1** | 2 | The forward error correction scheme. LDCP ½, 2/3, ¾  [0, 1, 2, 3] 🡪 [½, 2/3, ¾, ¾] Rate LDCP |
| **RM1** | 2 | The amount of rate matching (see table 1.1) |
| **BPS1** | 2 | The QAM constellation of each resource element.  [0, 1, 2, 3] 🡪 [BPSK, QPSK, 16-QAM, 64-QAM] |
| **EBS2** | 2 | The encoded block size (648, 1296, 1944 bits) |
| **NTB2** | 16 | The number of transport blocks to be sent in the payload. A value of 0 indicates that payload B is not transmitted. |
| **FEC2** | 2 | The forward error correction scheme. LDCP ½, 2/3, ¾  [0, 1, 2, 3] 🡪 [½, 2/3, ¾, ¾] Rate LDCP |
| **RM2** | 2 | The amount of rate matching (see table 1.1) |
| **BPS2** | 2 | The QAM constellation of each resource element.  [0, 1, 2, 3] 🡪 [BPSK, QPSK, 16-QAM, 64-QAM] |
| **User Bits** | 24 | User defined bits. |
| **CRC** | 16 | Cyclic Redundancy check bits that protect the 72 signal field bits |
|  | 88 |  |

Table ‑: Information Stored in the Signal Field

*Forward Error Correction in the Signal Field*

The 88 bits are then protected by a rate 1/3 (L = 7) convolutional encoder, and rate matching is then used to repeat the encoded bits until all BPSK / QPSK data symbols in the signal field’s OFDM symbols are completely filled.

## The Payload Fields

The payload is divided into two different fields, *A* and *B*, to give the modem the opportunity to transmit data with different amounts of protection or MCS (modulation and coding scheme). The data sent in each payload consists of a certain number of blocks, whose sizes are defined by the FEC encoding that is used.

*MAC Blocks*

A MAC block refers to the set of bits the media access controller must convey to the physical layer at any single time. There are nine different MAC block sizes defined as follows.

*Transport Blocks*

A transport block is a group of bits that can be conveniently encoded by the LDPC encoder. There are nine different transport blocks sizes, or *TBS*, which are illustrated in the table below. Because each block is protected by a 16-bit CRC, the number of bits, *MBS*, that the media access controller may convey per block is *TBS* – 16.

|  |  |
| --- | --- |
| **TBS** | **FEC - EBS** |
| **324** | LDPC ½ - 648 |
| **648** | LDPC ½ 1296 |
| **972** | LDPC ½ - 1944 |
| **432** | LDPC 2/3 - 648 |
| **864** | LDPC 2/3 - 1296 |
| **1296** | LDPC 2/3 - 1944 |
| **486** | LDPC ¾ - 648 |
| **972** | LDPC ¾ - 1296 |
| **1458** | LDPC ¾ - 1944 |

Table ‑: Legal Transport Block Sizes

*Encoded and Rate Matched Blocks*

The LDPC encoders produce encoded blocks of sizes of 648, 1296 and 1944. At that point, the bits may or may not be rate matched. During rate matching we will either repeat none of the bits, all of the bits once, all of the bits three times, or repeat all bits seven times. Therefore, a single encoded block will feature the following encoded block size, or *EBS*. The factor C1 may be 2, 3/2, or 4/3, and is used to compute the encoded block size, EBS, whereas the factor C2 may be 1, 2, 4, or 8, and is used to compute the rate-matched block size, *RBS*.

(after encoding)

(after rate matching)

The number of transmit blocks, *NTB*, will then determine the size of the payload as follows.

Each payload should be zero padded in order to fill up the last OFDM symbol.

# Preamble Definitions

## The AGC Burst

*Purpose*

The purpose of the AGC burst is to present the receiver with a waveform that features properties that allow for fast detection of the signal magnitude. As such, the waveform should be wideband and feature a low peak to average power ratio. The AgcBurst shall occupy 5 microseconds, which at 20.48MHz yields 102 samples. The AgcBurst is a Zadoff-Chu sequence defined as a time domain sequence that is generated via the following steps.

🡪 The derivation of the AgcBurst begins with the expression of a Zadoff-Chu sequence. The length of the sequence shall be *Nzc* = 887, the root index *u* = 34 and the discrete time index spans *n* = 0, 1, … *Nzc*-1.

🡪 We now take the *Nzc* = 887-point discrete Fourier transform.

By definition, we will have a certain number of positive frequency subcarriers, *m* = 0, 1, …*ScPositive*-1 and a certain number of negative frequency subcarriers, *m* = *Nzc - ScNegative*, … *Nzc*-1.

🡪 The FFT output shall be mapped into a length 1024 IFFT input bugger. The indexing into the input buffer is based on the tone convention.

The DC tone, *Tones*[0], remains unmapped as it will not survive the zero-IF down conversion process in the receiver.

🡪 Execute the IFFT and retain the first 102 samples.

## PreambleA

*Purpose*

*PreambleA* is used for two purposes.

🡪 It provides a signal that the baseband processor can use to easily detect the FlexLink packet at a SNR of -8dB.

🡪 It provides a signal that the baseband processor can use to calculate the average frequency offset generated either due to a mismatch in LO frequencies at the transmitter and receiver or due to Doppler shift.

*Duration*

The length of the *PreambleA* depend on whether the transmit terminal wants the receiver terminal to compute and remove the frequency offset. Computation of the frequency offset and reprogramming of the receive synthesizer should be completed by before the sample of PreambleB arrive at the receive antenna input. The duration of PreambleA for this scenario shall be 5120 samples, which at a clock rate of 20.48MHz yields 250 microseconds. For the case that frequency offset acquisition and removal are not requires, PreambleB shall feature a length of 512 samples or 25 microseconds.

*Definition*

The equation of the *PreambleA* is as follows.

The Python formulation below provides the definition for the discrete time implementation.

# --------------------------------------------------------------

# > GeneratePreambleA()

# --------------------------------------------------------------

def GeneratePreambleA(SampleRate: float = 20.48e6) -> np.ndarray:

    """

    This function generates the PreambleA Waveform

    """

    CosineFrequencyA = 4\*160e3

    CosineFrequencyB = 12\*160e3

    Ts           = 1/SampleRate

    NumSamples   = math.floor(220e-6 / Ts)

    Time  = np.arange(0, NumSamples\*Ts, Ts, dtype = np.float64)

    Tone1 = np.exp( 1j\*2\*np.pi\*CosineFrequencyA\*Time, dtype = np.complex64)

    Tone2 = np.exp( 1j\*2\*np.pi\*CosineFrequencyB\*Time, dtype = np.complex64)

    Tone3 = np.exp(-1j\*2\*np.pi\*CosineFrequencyA\*Time, dtype = np.complex64)

    Tone4 = np.exp(-1j\*2\*np.pi\*CosineFrequencyB\*Time, dtype = np.complex64)

    PreambleA = (1/4) \* (Tone1 + Tone2 + Tone3 + Tone4)

    return PreambleA, Time

### Packet Detection

The detection algorithm for the frequency offset is as follows. The function below does not return a value. The final implementation will make a decision based on the ratio between the covariance and variance estimates.

def DetectPreambleA(RxPreambleA:       np.ndarray

                  , SampleRate:        float = 20.48e6

                  , bShowPlot:         bool = False) -> float:

    # Error checking

    assert isinstance(RxPreambleA, np.ndarray)

    assert isinstance(SampleRate, float) or isinstance(SampleRate, int)

    assert isinstance(bShowPlot, bool)

    # Generate Halfband filter

    N    = 15;                                      # Number of taps

    n    = np.arange(0, N, 1, dtype = np.int32)

  Arg  = n/2 - (N-1)/4                           # Argument inside sinc function

    Hann = np.ones(N, np.float32) - np.cos(2\*np.pi\*(n+1)/(N+1)) # The Hanning window

    # Half Band Filter impulse response

    h   = np.sinc(Arg)\*Hann

    h   = h/sum(h)                                  # normalize to unity DC gain

    # Filter the RxPreambleA

    FilteredPreambleA = np.convolve(RxPreambleA, h)

    # Some definitions

    PeriodicityHz              = 4\*160e3

    PeriodSamples              = int(SampleRate / PeriodicityHz)

    RxLength                   = len(FilteredPreambleA)

    IntegrationLengthInSamples = 400

    CurrentCovariance    = np.zeros(RxLength - PeriodSamples, FilteredPreambleA.dtype)

    CurrentVariance      = np.ones (RxLength - PeriodSamples, FilteredPreambleA.dtype)

    Ratio                = np.zeros(RxLength - PeriodSamples, FilteredPreambleA.dtype)

    # Computing the covariance, variance and their ratio

    for Index in range(0, RxLength - PeriodSamples):

        A = FilteredPreambleA[Index + PeriodSamples]

        B = FilteredPreambleA[Index]

        CurrentCovariance[Index]     = CurrentCovariance[Index - 1] + A \* np.conj(B)

        CurrentVariance[Index]       = CurrentVariance[Index - 1]   + B \* np.conj(B)

        if Index >= IntegrationLengthInSamples:

            A = FilteredPreambleA[Index + PeriodSamples - IntegrationLengthInSamples]

            B = FilteredPreambleA[Index - IntegrationLengthInSamples]

            CurrentCovariance[Index]    -= A \* np.conj(B)

            CurrentVariance[Index]      -= B \* np.conj(B)

Ratio[Index] = np.abs(CurrentCovariance[Index])/ np.abs(CurrentVariance[Index])

        if Index < 100:

            Ratio[Index] = 0

    if bShowPlot == True:

        plt.figure(5)

        plt.subplot(3, 1, 1)

        plt.plot(np.arange(0, len(CurrentCovariance)), np.abs(CurrentCovariance))

        plt.title('The absolute value of the Covariance')

        plt.tight\_layout()

        plt.grid(True)

        plt.subplot(3, 1, 2)

        plt.plot(np.arange(0, len(CurrentVariance)), np.abs(CurrentVariance))

        plt.title('The absolute value of the Variance')

        plt.tight\_layout()

        plt.grid(True)

        plt.subplot(3, 1, 3)

        plt.plot(np.arange(0, len(Ratio)), Ratio)

        plt.title('The Ratio of absolute values of the Covariance and Variance')

        plt.grid(True)

        plt.tight\_layout()

        plt.show()

### Frequency Offset Detection

The frequency offset detection uses a 4096 FFT in order to determine the frequency offset.

# --------------------------------------------------------------

# > GeneratePreambleA()

# --------------------------------------------------------------

def ProcessPreambleA(RxPreambleA:       np.ndarray

                   , SampleRate:        float = 20.48e6

                   , bHighCinr:         bool  = False

                   , bShowPlots:        bool  = False) -> float:

    """

    This function estimates the frequency offset in the PreambleA Waveform

    """

    # ------------------------

    # Error checking

    # ------------------------

    FFT\_Size    = 4096

    assert np.issubdtype(RxPreambleA.dtype, np.complexfloating), 'Error.'

    assert len(RxPreambleA.shape) == 1,                          'Error.'

    assert len(RxPreambleA) >= FFT\_Size,                         'Error.'

    assert isinstance(bHighCinr, bool),                          'Error.'

    assert isinstance(bShowPlots, bool),                         'Error.'

    # ------------------------

    # Overlay the Hanning Window to induce more FFT leakage

    # ------------------------

    N              = FFT\_Size

    n              = np.arange(0, N, 1, dtype = np.int32)

    Hanning        = 0.5 - 0.5 \* np.cos(2\*np.pi \* (n + 1) / (N + 1))

    RxWaveform     = RxPreambleA[0:N].copy() \* Hanning

    # ------------------------------

    # Take the FFT and rearrange its output such that negative frequency bins are first

    # ------------------------------

    ZeroIndex      = int(FFT\_Size/2)

    FFT\_Output     = np.fft.fft(RxWaveform[0:FFT\_Size])

    FFT\_Rearranged = np.hstack([FFT\_Output[ZeroIndex : ], FFT\_Output[:ZeroIndex]])

    # ------------------------------

    # Find all peak bin indices

    # ------------------------------

    MaxIndex        = np.argmax(abs(FFT\_Rearranged))

    OffsetIndex     = MaxIndex - ZeroIndex

    # PeakIndexDeltas is the distance in bins of all peaks relative to the most

# negative peak index

    PeakIndexDeltas = np.array([0, 256, 512, 768])

    # Now that we have the peak, we need to figure out the indices of the other peaks

    if   OffsetIndex <  -320:  # Then the maximum peak is the one belonging to -1920MHz

            PeakIndices = MaxIndex + PeakIndexDeltas

    elif OffsetIndex >= -320 and OffsetIndex  < 0:   # MaxPeak at -640MHz

            PeakIndices = MaxIndex + PeakIndexDeltas - PeakIndexDeltas[1]

    elif OffsetIndex <=  320 and OffsetIndex >= 0:   # MaxPeak at  +640MHz

            PeakIndices = MaxIndex + PeakIndexDeltas - PeakIndexDeltas[2]

    elif OffsetIndex >   320:                        # MaxPeak at +1920MHz

            PeakIndices = MaxIndex + PeakIndexDeltas - PeakIndexDeltas[3]

    # ------------------------------

    # Use maximum ratio combining to compute the frequency offset

    # ------------------------------

    MRC\_Scaling = FFT\_Rearranged[PeakIndices] \* np.conj(FFT\_Rearranged[PeakIndices])

    Sum0 = np.sum(FFT\_Rearranged[PeakIndices - 3] \* MRC\_Scaling)

    Sum1 = np.sum(FFT\_Rearranged[PeakIndices - 2] \* MRC\_Scaling)

    Sum2 = np.sum(FFT\_Rearranged[PeakIndices - 1] \* MRC\_Scaling)

    Sum3 = np.sum(FFT\_Rearranged[PeakIndices + 0] \* MRC\_Scaling)

    Sum4 = np.sum(FFT\_Rearranged[PeakIndices + 1] \* MRC\_Scaling)

    Sum5 = np.sum(FFT\_Rearranged[PeakIndices + 2] \* MRC\_Scaling)

    Sum6 = np.sum(FFT\_Rearranged[PeakIndices + 3] \* MRC\_Scaling)

    # Note that the sample rate of 20.48MHz / 4096 results in tone spacing of 5KHz. If

# the frequency offset is beyond 5 KHz, then we must adjust the recreating

# frequencies below. If the frequency offset is less than 2.5KHz away, then the

# peaks will be at [1664, 1920, 2176, 2432]

    SubcarrierOffset = PeakIndices[0] - 1664

    n = np.array([(2048 - 256), (2048 + 256)])

    N = FFT\_Size

    if bHighCinr == False:

# Coarse reconstruction of waveform uses fewer FFT results

        Tone = np.exp( 1j\*2\*np.pi\*n\*(SubcarrierOffset-2)/N)   \* Sum1 + \

               np.exp( 1j\*2\*np.pi\*n\*(SubcarrierOffset-1)/N)   \* Sum2 + \

               np.exp( 1j\*2\*np.pi\*n\*(SubcarrierOffset  )/N)   \* Sum3 + \

               np.exp( 1j\*2\*np.pi\*n\*(SubcarrierOffset+1)/N)   \* Sum4 + \

               np.exp( 1j\*2\*np.pi\*n\*(SubcarrierOffset+2)/N)   \* Sum5

    else:

# Higher resolution reconstruction of waveform uses more FFT results

        Tone = np.exp( 1j\*2\*np.pi\*n\*(SubcarrierOffset-3)/N)   \* Sum0 + \

               np.exp( 1j\*2\*np.pi\*n\*(SubcarrierOffset-2)/N)   \* Sum1 + \

               np.exp( 1j\*2\*np.pi\*n\*(SubcarrierOffset-1)/N)   \* Sum2 + \

               np.exp( 1j\*2\*np.pi\*n\*(SubcarrierOffset  )/N)   \* Sum3 + \

               np.exp( 1j\*2\*np.pi\*n\*(SubcarrierOffset+1)/N)   \* Sum4 + \

               np.exp( 1j\*2\*np.pi\*n\*(SubcarrierOffset+2)/N)   \* Sum5 + \

               np.exp( 1j\*2\*np.pi\*n\*(SubcarrierOffset+3)/N)   \* Sum6

    Rotation1   = Tone[1] \* np.conj(Tone[0])

    AngleRad    = np.angle(Rotation1)

    AngleCycles = AngleRad / (2\*np.pi)

    FreqOffset  = AngleCycles \* SampleRate / 512

    if bShowPlots:

        print('Frequency Offset = ' + str(FreqOffset) + ' Hz')

        print('MaxIndex =    ' + str(MaxIndex))

        print('OffsetIndex = ' + str(OffsetIndex))

        print(PeakIndices)

        plt.figure(1)

        plt.stem(np.arange(0, len(FFT\_Rearranged)), np.abs(FFT\_Rearranged))

        plt.grid(True)

        plt.show()

    return FreqOffset

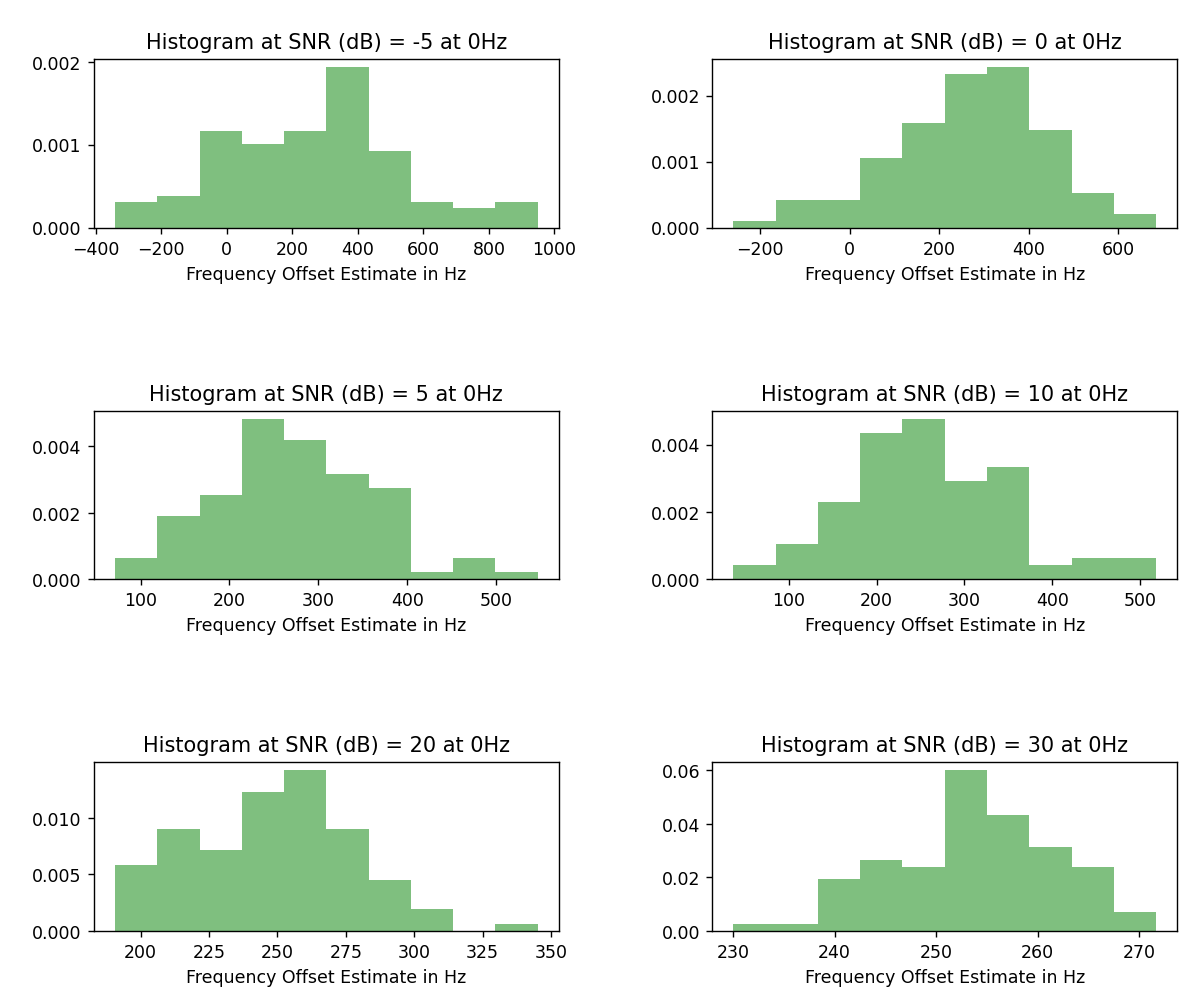


Figure ‑: Frequency Offset Estimation Performance for 250Hz Offset at Different SNR Conditions

## PreambleB

*Purpose*

The *PreambleB* is a sequence that facilitates timing synchronization and thereby the determination of the FFT timing instance in the receiver. A Zadoff-Chu sequence was chosen for this purpose as it features excellent autocorrelation properties, is easy to generate for different lengths and has a low peak to average power ratio. Whereas the low PAP feature is a great attribute, this advantage doesn’t necessarily contribute much to the transmit signal as OFDM has high a PAP to begin with. The low PAP feature brings much more to the table in signal carrier FDMA systems.

*Construction*

The Zadoff-Chu sequence will be defined as a time domain sequence that is generated via the following steps.

🡪 The derivation of the PreambleB begins with the expression of a Zadoff-Chu sequence. The length of the sequence shall be *Nzc* = 331, the root index *u* = 34 and the discrete time index spans *n* = 0, 1, … *Nzc*-1.

🡪 In order to map the sequence into the resource grid, we now take the *Nzc* = 331-point discrete Fourier transform.

By definition, we will have a certain number of positive frequency subcarriers, *m* = 0, 1, …*ScPositive*-1 and a certain number of negative frequency subcarriers, *m* = *Nzc - ScNegative*, … *Nzc*-1.

🡪 The FFT output shall be mapped into the resource grid according to the following rules. The indexing into the resource grid is based on the tone convention.

The DC tone, *Tones*[0], remains unmapped as it will not survive the zero-IF down conversion process in the receiver.

🡪 The *PreambleB* shall be OFDM modulated symbol given the mapping operation above. The code below assumes that the IFFT operation features length N = 1024.

% PreambleB Generation

Nzc = 331;

ScPositive = ceil(Nzc/2);

ScNegative = floor(Nzc/2);

u1 = 34;

n = 0:Nzc-1;

% Definition of the Zadoff-Chu Sequence

zc = exp(-1j\*pi\*u1\*n.\*(n+1)/Nzc);

% The Fourier Transform of the Zadoff-Chu Sequence

PreambleB\_FFT = (1/sqrt(Nzc))\*fft(zc);

% ------------------------------------

% Mapping into IFFT buffer and upsampling

% ------------------------------------

% Loading the PreambleB\_FFT sequence into the IFFT buffer (N=1024) for rendering into the time domain

IFFT\_Buffer1024 = zeros(1,1024);

IFFT\_Buffer1024(1, 1:ScPositive) = PreambleB\_FFT(1, 1:ScPositive);

IFFT\_Buffer1024(1, end-ScNegative+1:end) = PreambleB\_FFT(1, Nzc-ScNegative+1:Nzc);

IFFT\_Buffer1024(1, 1) = 0;

PreambleB = sqrt(1024)\*ifft(IFFT\_Buffer1024);



Figure ‑: Upsampling Process for Zadoff-Chu Sequence



Figure ‑: The Auto-correlation Performance of the PreambleB for N = 1024

# Transmitter Architecture

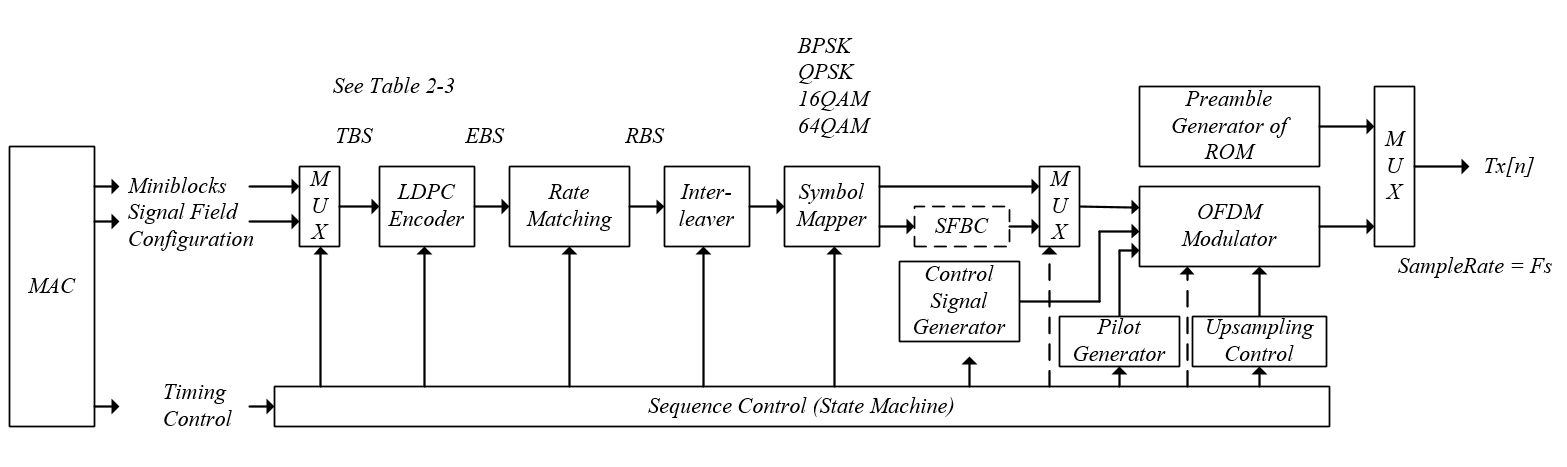


Figure ‑: Generic Transmitter Diagram

## Transmit Diversity

In this section, we discuss a variation of Alamouti’s space time encoder technique. This variation, called space frequency block coding, or SFBC, arranges successive symbols not along the time axis but along the frequency axis. The figure below illustrates our intent to transmit two symbols, *x*[0] and *x*[1], which we place in OFDM symbols 0 at successive subcarrier positions in the resource grid of transmit antenna 0. Notice the placement of the conjugated and negative conjugated symbols at the same positions in the resource grid of transmit antenna 1.



Figure ‑: Space Frequency Block Coding as Implemented in LTE

The data symbol stream *x*[*n*] that shall be mapped into a single OFDM symbol shall be mapped into the resource grid for transmit antenna 0 and transmit antenna 1 as shown in the figure above.

# Address Map

|  |  |  |  |
| --- | --- | --- | --- |
| **Addr** | **Information** | **Number of Bits / Position** | **Description** |
|  |  |  |  |
| **0x014** | **NTB1** | [15:0] | The number of transport blocks to be sent in the payload. A value of 0 is not valid. |
| **0x018** | **Payload1 Config** | [31:0] | contains signal field configuration information for payload one of the waveform |
|  |  | [1:0] | **EBS1:** The encoded block size (648, 1296, 1944 bits) |
|  |  | [5:4] | **FEC1:** The forward error correction scheme. LDCP ½, 2/3, ¾  [0, 1, 2, 3] 🡪 [½, 2/3, ¾, ¾] Rate LDCP |
|  |  | [9:8] | **RM1:** The amount of rate matching (see table 1.1) |
|  |  | [13:12] | **BPS1:** The QAM constellation of each resource element.  [0, 1, 2, 3] 🡪 [BPSK, QPSK, 16-QAM, 64-QAM] |
|  |  | [31:16] | **Reserved** |
|  |  |  |  |
| **0x024** | **NTB2** | [15:0] | The number of transport blocks to be sent in the payload. A value of 0 indicates that payload B is not transmitted. |
| **0x028** | **Payload2 Config** | [31:0] | contains signal field configuration information for payload two of the waveform |
|  |  | [1:0] | **EBS2:** The encoded block size (648, 1296, 1944 bits) |
|  |  | [5:4] | **FEC2:** The forward error correction scheme. LDCP ½, 2/3, ¾  [0, 1, 2, 3] 🡪 [½, 2/3, ¾, ¾] Rate LDCP |
|  |  | [9:8] | **RM2:** The amount of rate matching (see table 1.1) |
|  |  | [13:12] | **BPS2:** The QAM constellation of each resource element.  [0, 1, 2, 3] 🡪 [BPSK, QPSK, 16-QAM, 64-QAM] |
|  |  | [31:16] | **Reserved** |
| **0x1000 – 0x1FFF** | **payload 1 data** | [31:0] | Payload data for payload 1 is transmitted and received. |
|  |  |  |  |
| **0x2000 – 0x2FFF** | **payload 2 data** | [31:0] | Payload data for payload 2 is transmitted and received. |
|  | **Status** |  | under run  over run  empty  full  count  start  done  FIFO, or memory map? |
|  |  |  | signal strength: AGC burst final level (max amplitude)  snr rssi  packet arrived,  packet detected |

# OFDM comparisons

## Comparison table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 802.11a / c / x | 802.16 | DVB-T2 | LTE 4G / 5G |
| Freq bands: | 4.9-6.1GHz  2.4,5GHz |  |  |  |
| Bandwidth | 20,40,80… MHz | 10,20 MHz | 8,10MHz | 1.25; 20MHz |
| occupied bandwidth |  |  |  | 1.140, 18.015MHz |
| Sampling Freq |  | 11.2, 22.4MHz |  |  |
| subcarrier spacing (df) | a: 20MHz/64 = 312.5kHz  ac:312.5kHz  ax:78.125kHz | 10.9375, 10.9375KHz | 2K: 4,464Hz  8K: 1,116Hz  279–8,929Hz | 4G: 30.73MHz / 2048 = 15kHz,  5G: 30kHz |
| used symbol length | a: 3.2us  ac: 3.2us  ax: 12.8us | 91.429us (Tu) | 2k: 224us  8k: 896us  112–3,584us | 66.67us (15kHz)  33.33us (30kHz) |
| cp | a: 0.8us (239.8m)  ac:0.8/0.4  ax: 0.8/1.6/3.2 | Tg= 1/8Tu,1/16Tu,1/4Tu | 2k: 56us (1/4) | 15kHz: 5.2/4.69us short  16.67us long  30kHz: 2.34us |
| cp distance | a:  ac:  ax: |  |  |  |
| Symbol length  (cp+symbol) | ac:0.8/0.4 + 3.2us  ax: 0.8/1.6/3.2 + 12.8 us | 102.857us,  97.143us,  114.853us, |  | 71.35 (15kHz),  35.68 (30kHz) |
| GI (fraction) |  | Tg= 1/8Tu,1/16Tu,1/4Tu | 1/4, 19/128, 1/8, 19/256, 1/16, 1/32, 1/128 |  |
| FFT | a:64  ac:  ax: | 1024, 2048 | 1k,2k, 4k, 8k, 16k.. | 2048; 4096 |
| Active subcarriers | a: 52 |  | 853–27,841  1705(2K)  6817(8K) | 76;1200 |
| guard subcarriers |  |  |  | 52; 847 |
| Modulations | QPSK – 256QAM,  QPSK – 1024QAM |  | QPSK, 16QAM, 256QAM | QPSK – 256QAM |
| FEC | a: convolutional coding (1/2,2/3,3/4) |  | inner:LDPC  1⁄2, 3⁄5, 2⁄3, 3⁄4, 4⁄5, 5⁄6  outer:BCH;  1/2, 3/5, 2/3, 3/4, 4/5, 5/6, 6/7, 8/9 |  |
| time interleaving |  |  | up to 250ms |  |

# Appendix

Links: