MANUAL

LabVIEW Communications LTE Application Framework 19.5

This document provides detailed, technical information about how to use LabVIEW Communications LTE Application Framework.

Contents

1 Introduction 2

2 Scope 3

2.1 Operation Modes 4

2.2 PHY 5

2.2.1 Frame Structure, Bandwidth Mode, CP Mode and Physical Resource Grid 5

2.2.2 PHY DL Channels and Signals 7

2.2.3 Physical UL Channels and Signals 16

2.3 MAC for Windows Host and FPGA 21

2.3.1 System Configuration 21

2.3.2 DL Scheduling 21

2.3.3 UL Scheduling 22

2.3.4 DL MAC Packet for User-Defined Data 23

2.3.5 UL MAC Packet for Feedback Information 23

2.4 Interfacing PHY and MAC on Real-Time Targets 26

2.4.1 PHY SAP 26

2.4.2 Confirmation Handling 28

3 Implementation Details 32

3.1 Architecture 32

3.2 FPGA Implementation 34

3.2.1 RF Interface 35

3.2.2 DL TX 36

3.2.3 DL RX 39

3.2.4 UL Transmitter 46

3.2.5 UL Receiver 46

3.2.6 Clocking Considerations 47

3.3 Host Implementation 48

NATIONAL INSTRUMENTS

3.3.1 Initialization, Synchronize Exit Condition and Cleanup 50
3.3.2 Configure RX/TX Baseband and RF 51
3.3.3 Synchronization and Automatic Gain Control 51
3.3.4 Update Graphs and Indicators 51
3.3.5 Compute Throughput and Block Error Rate 51
3.3.6 Update Dynamic DL Configuration (Includes Rate Adaptation) 52
3.3.7 Receive UDP Data and Send UDP Data 54
3.3.8 Select Constellation to Display 54
3.4 Real-Time (RT) Implementation Overview 54
3.4.1 Transport Layer 54
3.4.2 Data Flow 55
3.4.3 Data Handling Loops 56
4 Conclusion 57
5 Appendix 58
5.1 LTE Application Framework PHY SAP Message Definitions 58
5.1.1 Message Numbering Schema 58
5.1.2 Used Message IDs 59
5.1.3 General Message Header (Applies to All Messages) 60
5.1.4 SAP Sub-header 61
5.1.5 Messages 61
5.2 Component and Namespace Layout 74
5.2.1 Component Layout 74
5.2.2 Namespacing 74
5.2.3 Extension of the Framework 75
6 Abbreviations 75
7 Bibliography 77

1 Introduction

LTE Application Framework provides a ready-to-run, easily modifiable real-time physical layer (PHY) and lower medium access control (MAC)-layer reference design based on the long-term evolution (LTE) wireless standard. The application framework is available with LabVIEW Communications System Design Suite (LabVIEW Communications).

This application framework provides a substantial starting point for researchers looking for ways to improve the LTE standard by exploring brand-new algorithms and

architectures that can support the tremendous increase of the number of terminals, inventing new waveforms by which to modulate and demodulate the signals, or finding new multi-antenna architectures that fully exploit the degrees of freedom in the wireless medium.

The application framework is comprised of modular PHY and MAC blocks implemented using LabVIEW Communications. It is designed to run on the powerful Xilinx Kintex-7 FPGA and an Intel x64 general purpose processor, which are tightly integrated with the RF and analog front ends of the NI software defined radio (SDR) hardware.

The framework is designed from the ground up for easy modifiability, while adhering to the main specifications of the LTE standard. This design allows wireless researchers to quickly get their real-time prototyping laboratory set up and running based on the LTE standard. They can then primarily focus on selected aspects of the protocol that they wish to improve, and easily modify the design and compare their innovations with the existing standards.

2 Scope

The application framework provides the functional elements of the PHY layer as well as the MAC layer of both base station (eNodeB) and user equipment (UE). This code includes the following elements:

- Downlink (DL) transmission (TX) and reception (RX)
- Uplink (UL) TX and RX

Additionally, basic MAC functionalities are provided which allow for the following features:

- Packet-based user data transmission in DL, enabling user data streaming applications
- Feedback of DL channel state information and DL (hybrid automatic repeat request (HARQ)) acknowledgment (ACK)/negative acknowledgement (NACK) through the UL
- Basic adaptive modulation and coding (AMC), which includes link adaptation in DL, enabling DL closed loop operations

The following subsections describe in more detail which principal operation modes are provided by the application framework and which specific subset of PHY and MAC functionalities of a 3rd Generation Partnership Project (3GPP) LTE release 10 compliant system is implemented.

Deviations and simplifications with respect to the 3GPP LTE release 10 standard are also described here. They have been applied to keep the complexity of the application framework at a reasonable level.

2.1 Operation Modes

The application framework offers three operation modes, as depicted in Figure 1. These three operation modes are provided by the following pairs of top-level host and FPGA implementations:

- **Downlink**
  - Can be used to establish a DL in either a single-device setup or a double-device setup.
  - Implements the DL TX of an eNodeB and the DL RX of a UE including the basic DL TX and DL RX MAC functionalities
  - In a single-device setup, a special *MAC shortcut* (refer to Figure 18 in section **Error! Reference source not found.**) allows for DL AMC (rate adaptation) even without a real UL feedback channel
  - Top-level host VI: LTE Host DL.gvi
  - The top-level FPGA VI is one of the following VIs:
    - LTE FPGA FlexRIO DL.gvi
    - LTE FPGA USRP RIO 40 MHz BW DL.gvi
    - LTE FPGA USRP RIO 120 MHz BW DL.gvi
- **eNodeB**
  - Provides the eNodeB side in a double-device setup.
  - Implements the DL TX and the UL RX of an eNodeB including the basic eNB MAC functionalities (refer to the upper part of Figure 19)
  - Top-level host VI: LTE Host eNodeB.gvi
  - The top-level FPGA VI is one of the following VIs:
    - LTE FPGA FlexRIO eNodeB.gvi
    - LTE FPGA USRP RIO 40 MHz BW eNodeB.gvi
    - LTE FPGA USRP RIO 120 MHz BW eNodeB.gvi
- **UE**
  - Provides the UE side in a double-device setup
  - Implements the DL RX and the UL TX of a UE including the basic UE MAC functionalities (refer to the lower part of Figure 19)
  - Top-level host VI: LTE Host UE.gvi
  - The top-level FPGA VI is one of the following VIs:
    - LTE FPGA FlexRIO UE.gvi
    - LTE FPGA USRP RIO 40 MHz BW UE.gvi
    - LTE FPGA USRP RIO 120 MHz BW UE.gvi

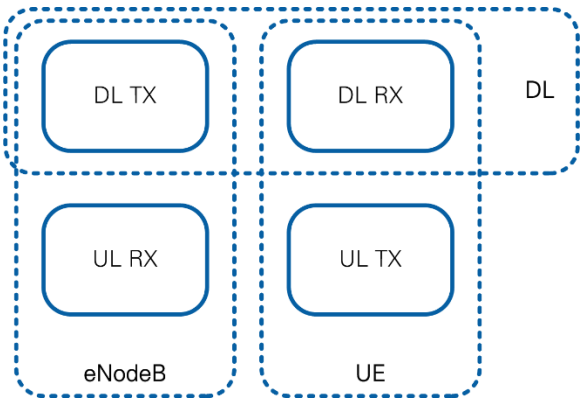

Figure 1: System Configurations (Host and Associated FPGA Code)

The DL operation mode can be used either in a single-device setup or in a double-device setup. The eNodeB/UE operation modes require a double-device setup.

2.2 PHY

The application framework implements parts of a 3GPP-LTE release 10 compliant DL and UL PHY TX and RX. To keep the complexity of this application framework at a reasonably low level, only a subset of the physical layer features defined for 3GPP-LTE release 10 compliant devices is implemented. This subset also includes feature simplifications and limitations of the configurability of the implemented PHY features, for example, specific configuration parameters are fixed to single values, and others are only quasi-statically configurable. Notice that fixed parameter settings can only be changed by modifying the design.

The following subsections give a detailed overview over the implemented PHY features, used simplifications, and (potentially restricted) PHY configuration capabilities with respect to the 3GPP LTE releases 10 specifications.

2.2.1 Frame Structure, Bandwidth Mode, CP Mode and Physical Resource Grid

The application framework supports the following (partially fixed) configurations:

- Bandwidth mode: 20 MHz (100 physical resource blocks (PRBs))
- Cyclic prefix (CP) configuration: Normal cyclic prefix
- Frame Structure:
    - **Type 1**—Frame structure type 1 (FDD)
    - **Type 2**—Frame structure type 2 (TDD)
        - TDD UL-DL configuration: 5
        - Special subframe configuration: 5

The detailed radio frame structure for both frame structure types is shown in Figure 2. Each radio frame is 10 ms long and consists of 10 subframes. Each subframe has a length of 1 ms, which comprises 30,720 complex time-domain baseband samples sampled at a rate of 30.72 MS/s, which is valid for the 20 MHz LTE bandwidth mode. The related sample period *Ts* is (1/30.72e6) s. The types of subframes vary with the subframe index in dependence on the selected radio frame type. DL subframes (D) are reserved for DL transmissions; UL subframes (U) are reserved for UL transmissions. Special subframes (S) are used with TDD only. For TDD UL-DL configuration 5, which is supported by the application framework, there is only one special subframe per radio frame. Special subframes consists of the following fields:

- **DwPTS**—DL pilot time slot
    - Reserved for DL transmission

- In the application framework, it is restricted to transmission of PHY DL control channel (PDCCH) and cell-specific reference signals (CRS)
- The length is fixed to 3 orthogonal frequency-division multiplexing (OFDM) symbols (6,592 × *Ts*)

- **GP**—Guard Period
  - Time-domain guard period for switching between active DL transmission/reception and active UL RX/TX
  - The length is fixed to 9 OFDM symbols (19,744 × *Ts*)
- **UpPTS**—UL pilot time slot
  - Reserved for transmission of UL sounding reference symbols (SRS)
  - The length is fixed to 2 OFDM symbols (4,384 × *Ts*)

According to *TS36.211: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 10)* [1], the length of the special subframe fields varies with the special subframe configuration and the selected cyclic prefix configuration. In the current application framework, implementation of both parameters are fixed to specific settings, so the special subframe fields have a fixed length.

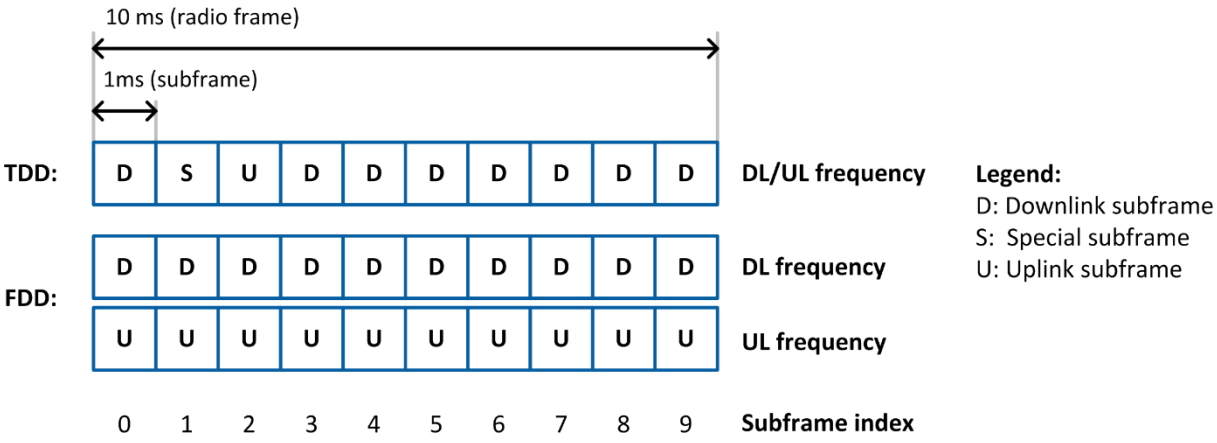

Figure 2: Subframe Types for TDD and FDD Frame Structure

For normal cyclic prefix configuration, each subframe is divided into 14 OFDM symbols. The OFDM symbol duration is 2,048 × *Ts* extended by a cyclic prefix of 160 × *Ts* for OFDM symbols 0 and 7 and a cyclic prefix of 144 × *Ts* for all other OFDM symbols in a subframe.

For the 20 MHz LTE bandwidth mode, a 2,048-point inverse fast Fourier transform (IFFT) is specified to be used in the OFDM modulator; in other words, 2,048 frequency-domain subcarriers per OFDM symbol are available. According to *TS36.211: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 10)* [1], only the inner 1,200 subcarriers, excluding the DC carrier, are allowed to be used for actual signal transmissions. The remaining unused subcarriers act as guard band to neighboring channels. The set of 1,200 usable

subcarriers, also called resource elements, are organized in sets of 12 contiguous subcarriers corresponding to the PRB. Notice that one PRB comprises the same set of 12 contiguous subcarriers, or resource elements, of multiple consecutive OFDM symbols, such as all OFDM symbols of a slot or subframe. For further details, refer to *TS36.211: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 10)* [1].

2.2.2 PHY DL Channels and Signals

The DL TX and DL RX each include FPGA implementations of the following channels:

- Primary synchronization signal (PSS)
- CRS
- UE-specific reference signals (UERS)
- PDCCH
- Physical DL shared channel (PDSCH)

The following DL channels and signals are not implemented:

- Secondary synchronization signal (SSS)
- Multimedia broadcast multicast service single frequency network (MBSFN) reference signals
- Positioning reference signals
- Channel state information (CSI) reference signals
- Physical control format indicator channel (PHICH)
- Physical HARQ indicator channel (PCFICH)
- Physical broadcast channel (PBCH)
- Physical multicast channel (PMCH)

The supported physical channels and signals are generally implemented in compliance with 3GPP LTE release 10 specifications as described in *TS36.211: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 10)* [1], *TS36.212: Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding (Release 10)* [2], and *TS36.213: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (Release 10), V10.12.0* [3]. Any specific deviations, extensions, simplifications, or configuration restrictions are explained in the corresponding subsections below.

A general restriction, which holds for all channels and signals if not mentioned otherwise, is that the Cell-ID is fixed to 0.

2.2.2.1 PSS

The PSS is transmitted only once per radio frame, with a periodicity of 10 ms instead of a periodicity of 5 ms. This adaption to the LTE specification is necessary to realize

a unique detection of the radio frame start without SSS support. Depending on the selected frame structure type the PSS is transmitted in one of the following ways:

- The third OFDM symbol of subframe 1 for TDD
- The seventh OFDM symbol of subframe 0 for FDD

The PSS sequence generation is implemented for Cell-ID 0 only.

2.2.2.2 CRS

CRS resource elements are always reserved (allocated) for two antenna ports. Active CRS transmissions are done on the first antenna port only.

Sequence generation as well as resource mapping (cell-specific frequency shift) are implemented for the fixed Cell-ID 0 only.

2.2.2.3 UERS

UERS can be optionally enabled in additional to CRS. The following features and configurations are supported:

- Supported on antenna ports 7 to 14; not supported on antenna port 5.
- Supported in DL subframes only; not supported in special subframes
- Sequence generation as specified in *TS36.211: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 10)* [1] for antenna ports 7 to 14.
  - Implemented for fixed Cell-ID and n_{SCID} parameter fixed to 0.
- With UERS support enabled, the resource elements for both UERS antenna port sets {AP 7, 8, 11, 12} and {AP 9, 10, 13, 14} are reserved in each slot of a DL subframe while active UERS transmission and reception is only performed for the selected antenna port. This is a deviation from (an extension to) the 3GPP LTE specification described in *TS36.211: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 10)* [1] and would be a prerequisite for TM9, like multiuser multiple input, multiple output (MIMO) for up to 8 UEs.

If UERS are enabled, UERS-based channel estimates, instead of CRS-based channel estimates, are used by the DL RX to equalize the PDSCH.

Note: The UERS-based channel estimation in the DL RX applies a simple interference cancellation scheme to reduce interference potentially caused by simultaneous code-orthogonal UERS transmissions on the same UERS resource elements. The underlying cancellation algorithm, which cannot be disabled in the current implementation, is designed for slowly time-varying radio channels only,

which means that it assumes the radio channel to be sufficiently time-invariant over a time period of one subframe (1 ms).

2.2.2.4 PDCCH

The PDCCH implemented in the application framework mainly follows the specifications described in *TS36.211: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 10)* [1], *TS36.212: Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding (Release 10)* [2], and *TS36.213: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (Release 10), V10.12.0* [3], but it is partially simplified and therefore partially proprietary.

2.2.2.4.1 PDCCH Resource Grid

The application framework supports the PDCCH to be transmitted only in the first OFDM symbol of each DL or special subframe. This corresponds to a fixed control format indicator (CFI) of 1. All available (non-CRS occupied or reserved) resources are used for PDCCH as PCFICH and PHICH are not part of the implementation.

2.2.2.4.2 PDCCH Format

The LTE standard allows for several different PDCCH formats (refer to section 6.8.1 of *TS36.211: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 10)* [1]), which differ in the so-called aggregation level, such as the number of consecutive control channel elements (CCEs) used for transmitting one DL control information (DCI) message on the PDCCH. One CCE comprises nine resource element (RE) groups (REGs) with four REs per REG. Since the PDCCH is specified to use quadrature phase-shift keying (QPSK) modulation, 2 bits per RE can be transmitted, which is 72 bits per CCE.

The PDCCH format implemented in the application framework is fixed to format 1, which corresponds to a fixed aggregation level of 2. Thus, always 2 consecutive CCEs are used for transmitting one DCI message. This format corresponds to the allocation of 18 REGs (72 REs), which allows for the transmission of 144 encoded bits per DCI message.

2.2.2.4.3 DCI Format

The implemented DCI format is partially proprietary. It corresponds to a subset of the LTE DCI Format 1. As illustrated in Figure 3, it consists of the following three fields:

- PRB allocation—for signaling the PRB allocation for the PDSCH
- Modulation coding scheme (MCS)—for signaling the PDSCH modulation and coding scheme
- TPC—reserved, for example, for UL TX power control commands (not currently used)

Each bit of the PRB Allocation field represents 4 PRBs (according to DL resource allocation type 0. For more information, refer to section 7.1.6.1 of *TS36.213: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (Release 10), V10.12.0* [3]). The leftmost bit represents the lowest resource block (group) index.

The MCS signaling is compliant with the LTE standard. For more information, refer to section 7.1.7 of *TS36.213: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (Release 10), V10.12.0* [3]. The supported MCS value range is restricted to 0...28. MCS 29, 30 and 31 are not supported as there is no HARQ processing included.

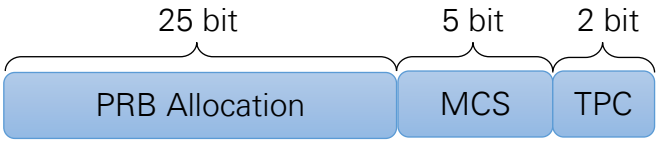

Figure 3: PDCCH DCI Format

Note: With the described DCI format, only DL scheduling grants can be signalized using the PDCCH, while UL scheduling information cannot be sent with PDCCH. Thus, the UL TX at the UE, as well as the UL RX at the eNB, must be configured manually using the appropriate host application.

Thirty-two overall bits are used for the supported DCI format.

2.2.2.4.4 DCI Encoding

The DCI encoding mainly follows the 3GPP LTE release 10 specification. Cyclic redundancy check (CRC) attachment and channel coding are fully compliant with *TS36.212: Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding (Release 10)* [2], but the rate matching is substituted by a simple parallel-to-serial conversion as is shown in Figure 4. According to *TS36.212: Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding (Release 10)* [2], the LTE rate-matching block requires one subblock interleaver per parity bit output stream of the convolutional channel encoder, followed by a bit collection stage and a circular buffer, which is used for the actual rate adaptation (that is, for bit puncturing (pruning) or repetition).

For the combination of the fixed PDCCH format (section 2.2.2.4.2) and the fixed modified DCI format (section 2.2.2.4.3) implemented by the application framework, there is no need for any rate adaptation. The number of output bits delivered by the 1/3-rate convolutional channel encoder matches the number of bits that can be transmitted by the supported PDCCH format 1 (2 CCEs → 144 bits). Thus, the rate matching can be skipped in this case.

Note: For reasons of simplicity, the parallel-to-serial conversion implemented in the application framework instead of the rate matching collects the parity output bits of the convolutional encoder as follows:

$$d_0^{(0)}, d_0^{(1)}, d_0^{(2)}, d_1^{(0)} d_1^{(1)}, d_1^{(2)}, \ldots$$

This equation is different than the way the bit collection stage is defined for the LTE-compliant rate matching. This stage collects the bits as follows:

$$v_0^{(0)}, v_1^{(0)}, \ldots, v_{K_\Pi}^{(0)}, v_0^{(1)}, v_1^{(1)}, \ldots, v_{K_\Pi}^{(1)}, v_0^{(2)}, v_1^{(2)}, \ldots, v_{K_\Pi}^{(2)}$$

where $v^{(i)}$ stands for the interleaved version of i^{th} parity bit output stream $d^{(i)}$ of the channel encoder.

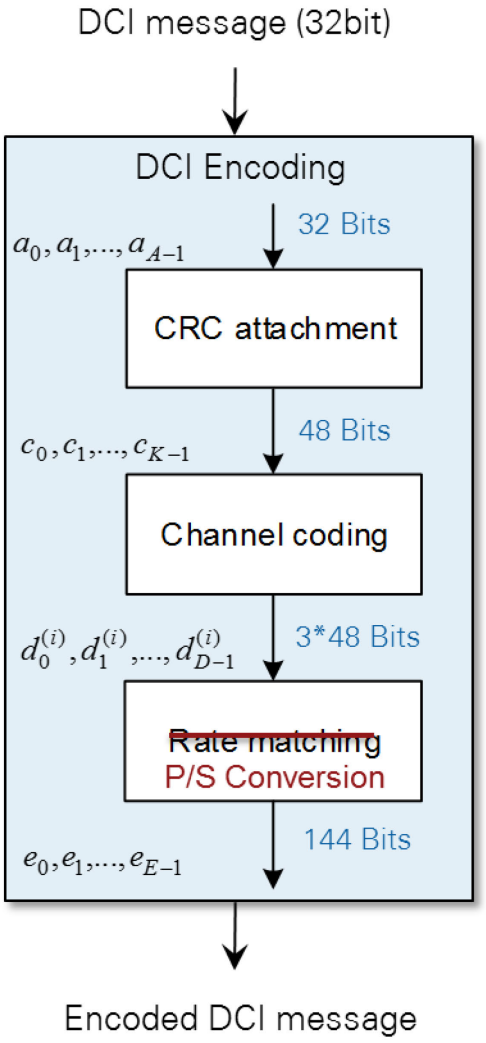

Figure 4: Modified DCI Encoding Scheme

2.2.2.4.5 PDCCH Multiplexing, Scrambling, Modulation, and Resource Mapping

Figure 5 shows further PDCCH processing with the following functionalities:

- PDCCH multiplexing
- Scrambling
- Modulation
- Layer mapping and precoding
- Mapping to resource elements

It is implemented in compliance with the specifications described in *TS36.211: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 10)* [1], with the following simplifications:

- PDCCH multiplexing currently supports only one control channel (DCI message) to be transmitted per subframe. There are eight possible CCE positions, which are pairs of consecutive CCEs, that can be manually selected through the CCE offset parameter, a prerequisite for multi-user support for up

to eight UEs. There is also no support of automatic CCE selection (PDCCH assignment) in dependence on radio network temporary identifier (RNTI) and search space (as defined in section 9.1.1 of *TS36.213: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (Release 10), V10.12.0* [3]). There is no need of blind PDCCH decoding procedures in the DL RX.

- QPSK modulation as specified in *TS36.211: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 10)* [1], but for reasons of implementation efficiency, modulation is performed after the REG interleaving.
  Note: This specific implementation does not influence the final output of the overall PDCCH processing.
- Layer mapping and precoding are only supported for transmission on a single antenna port (transmission mode 1). Since this is a one-to-one mapping, the block must not be implemented at all. REG interleaving is supported for fixed Cell-ID value 0 only. For this Cell-ID value, the cell-specific cyclic shift, which is specified in *TS36.211: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 10)* [1], in addition to the actual interleaver functionality, becomes transparent and therefore is not implemented.

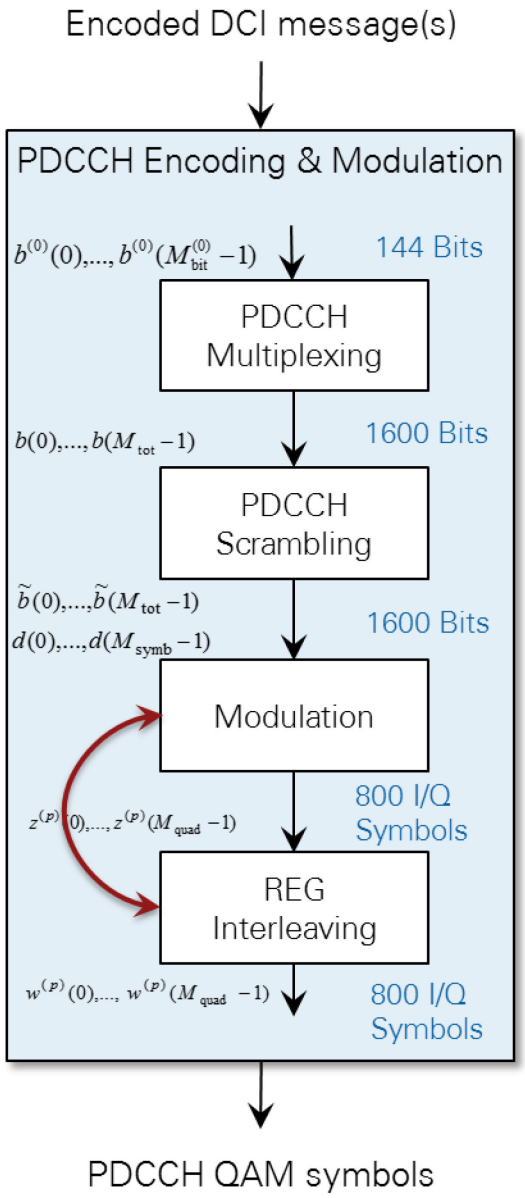

Figure 5: PDCCH Processing

2.2.2.5 PDSCH

Channel coding, scrambling, and modulation of the PDSCH are implemented in the application framework in compliance with *TS36.211: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 10)* [1] and *TS36.212: Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding (Release 10)* [2] with the following restriction: HARQ support is not implemented.

Layer mapping and precoding are only implemented for transmission on single antenna port (transmission mode 1).

The PDSCH resource mapping is compliant with *TS36.211: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 10)* [1], with the following exceptions:

- PDSCH transmission is supported in DL subframes only; it is not supported in special subframes.
- The PDSCH resource mapping is adapted to the modified or restricted resource mapping of PSS, CRS, and UERS.
- Resource elements outside the PDCCH region available from channels and signals that are not implemented (for example, SSS, PBCH) are used for PDSCH.
- For activated UERS, PDSCH is not transmitted in resource blocks (RBs) in which parts of the UERS (due to potential overlap with other PHY channels or signals) are not transmitted.
    - This restriction currently applies for FDD where the PRBs centered around DC in subframe 0 are not used for PDSCH transmissions due to the potential overlap with the PSS.
    - To ease the handling for the user, the related conditions will be automatically checked by the FPGA implementation of the DL TX and the PDSCH resource allocation will be automatically modified for the affected subframes. The modified PDSCH resource allocation will be applied for the whole PDSCH processing (including the transport block size determination) as well as for the related DCI content signaled on the PDCCH. As the currently implemented DCI format only allows for signaling the allocation of 4 consecutive PRBs, the 12 innermost PRBs (not only the 6 PRBs colliding with PSS) will not be used for PDSCH transmission in this special case.

The application framework supports a quasi-static PDSCH resource allocation at the DL TX using resource allocation type 0 according to *TS36.213: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (Release 10), V10.12.0* [3]. Thus, for the supported 20 MHz bandwidth mode, 25 resource block groups (RBGs) can be individually allocated. Each RBG addresses a set of 4 consecutive PRBs.

The PDSCH modulation order and transport block size determination follows the specifications in tables 7.1.7.1-1 and 7.1.7.2.1-1 of *TS36.213: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (Release 10), V10.12.0* [3]. They can be controlled at the DL TX by means of the MCS parameter. The application framework supports all MCSs between 0 and 28. MCS 29…31, which are only applicable with full DL HARQ support, are not implemented. For activated UERS MCS 28 is not supported since the resulting code rate would exceed 1 due to the modified UERS mapping.

Note: The selected PDSCH resource allocation as well as the selected MCS are signaled to the DL RX through the PDCCH. Thus, the PDSCH configuration is applied automatically in the receiver and has not to be set manually.

2.2.2.6 Supported Downlink Resource Grid

Figure 6 shows the resulting LTE DL resource grid for TDD with all supported physical DL channels and signals. It exemplarily shows the resource grid for the inner PRBs centered around DC which contain PSS. The resource mapping in the outer PRBs is in principle the same with the only difference that they do not contain PSS, instead the corresponding resource elements are left blank.

Figure 7 shows the resulting LTE DL resource grid for FDD with all supported physical DL channels and signals. The resource grid for both kind of PRBs, that is, inner PRBs (containing PSS and PSS reserved REs) and outer PRBs (not containing PSS) are shown.

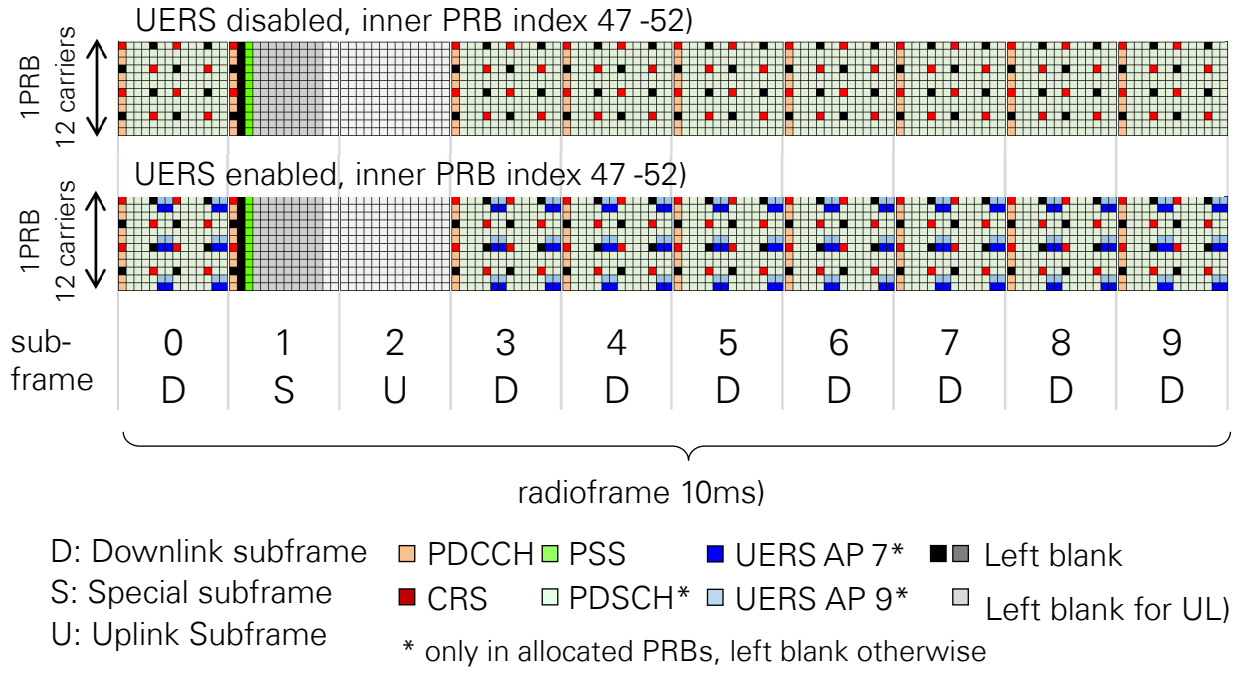

Figure 6: Supported LTE DL Resource Grid for TDD

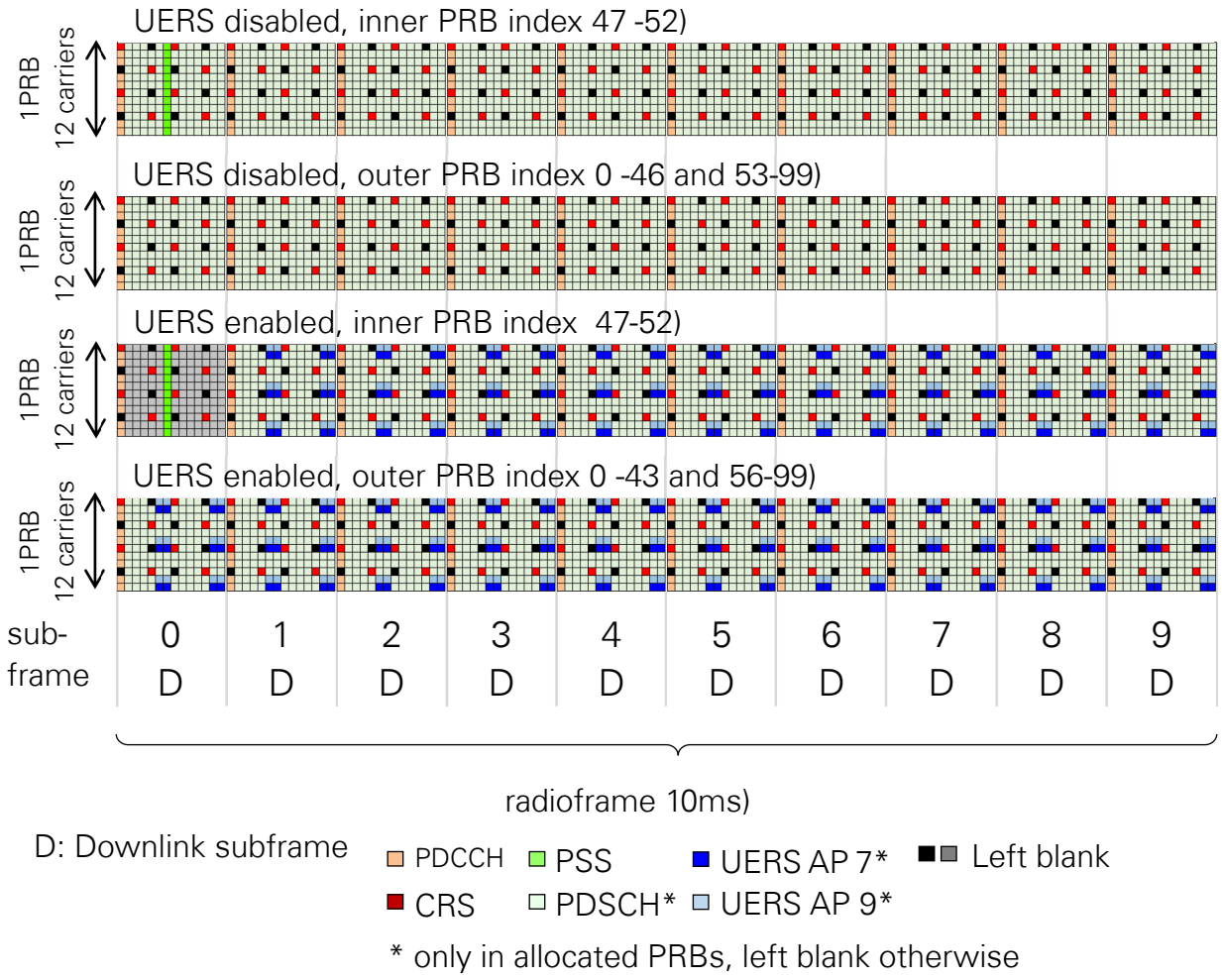

Figure 7 Supported LTE DL Resource Grid for FDD

2.2.3 Physical UL Channels and Signals

The UL implementation in the application framework is using OFDM access (OFDMA) instead of single-carrier frequency-division multiplexing access (SC-FDMA). Neither the physical UL shared channel (PUSCH) discrete Fourier transform (DFT) spreading nor the LTE UL specific half subcarrier shift are applied. Simple OFDM modulation is used instead with a zero DC subcarrier inserted, similar to the LTE DL specification.

The UL TX and RX implementations comprise the following physical channels and signals:

- PUSCH
- Demodulation reference signals (DMRS)
- SRS

The following UL channels and signals are not available:

- Physical random access channel (PRACH)
- Physical UL control channel (PUCCH)

2.2.3.1 PUSCH

The PUSCH implemented in the application framework applies a slightly simplified coding scheme in comparison to the specifications in *TS36.212: Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding (Release 10)* [2]. The applied simplifications are illustrated in Figure 8. The resulting coding scheme is very similar to the coding scheme used for PDSCH. The specifics in comparison to full standard compliant PUSCH encoding are the following:

- Only the coding for the UL shared channel (UL-SCH) data is supported
- Coding and multiplexing of UL control information (UCI) on PUSCH is not supported

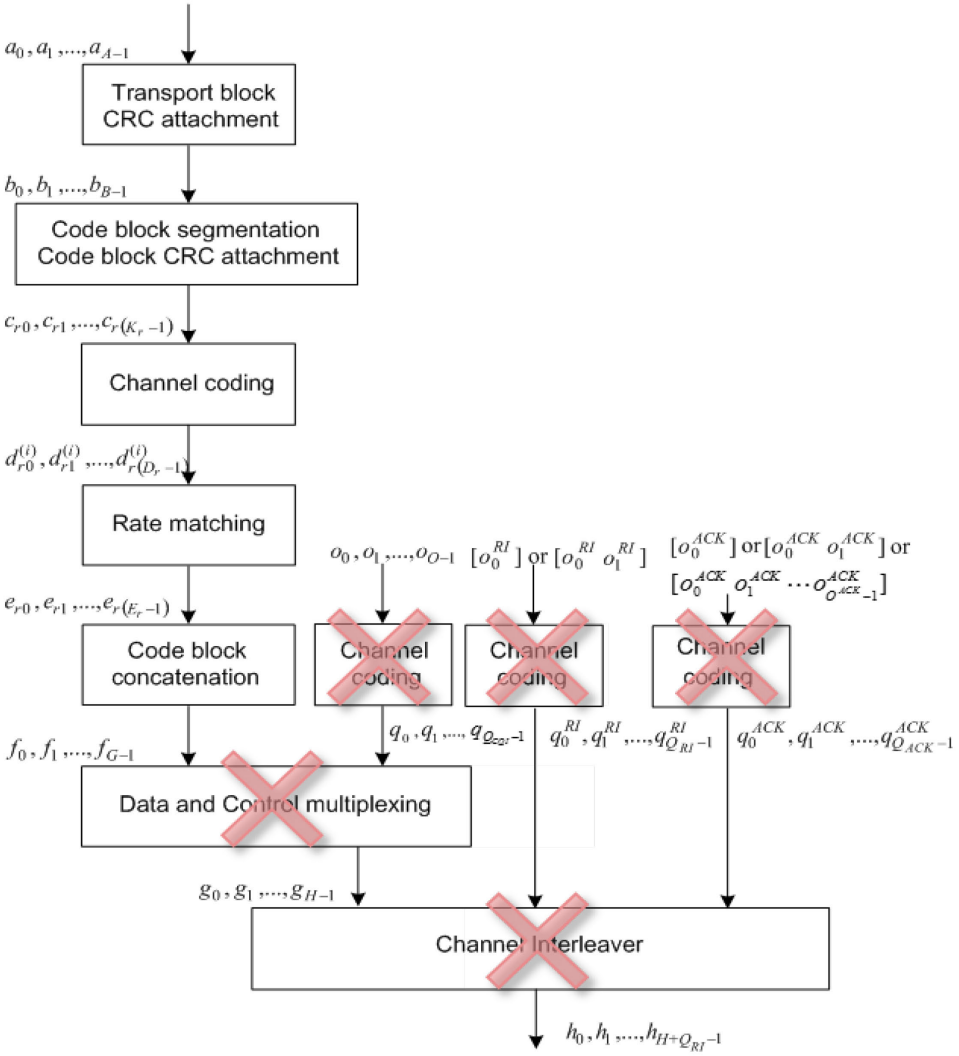

Figure 8: Simplified UL-SCH Coding Scheme

PUSCH scrambling, modulation, layer mapping, and pre-coding are implemented in compliance with *TS36.211: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 10)* [1] with the following restriction: layer mapping and precoding are implemented for transmission on single antenna port only.

The PUSCH transform precoding (DFT spreading) specified in section 5.3.3 of *TS36.211: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 10)* [1] is not implemented since the application framework applies OFDM instead of SC-FDMA in the UL.

The supported PUSCH resource mapping is compliant with the specifications in *TS36.211: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 10)* [1] with the following restrictions:

- For activated SRS support, the last OFDM symbol cannot be used for PUSCH at all, even if the UE is not actively transmitting SRS.
- PUSCH transmissions are only allowed on contiguous resource block sets.
- UL frequency hopping (PUSCH hopping) is not implemented.

The application framework supports a quasi-static PUSCH resource allocation at the UL TX and UL RX using resource allocation type 0 according to *TS36.213: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (Release 10), V10.12.0* [3]. Similar to the PDSCH allocation in the DL, 25 RBGs can be individually allocated for PUSCH in UL. Each RBG addresses a set of four consecutive PUSCH PRBs.

In the application framework, the PUSCH modulation order and transport block size determination uses the same MCS tables (tables 7.1.7.1-1 and 7.1.7.2.1-1 of *TS36.213: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (Release 10), V10.12.0* [3]) as applied for the PDSCH. This is a slight deviation from the LTE standard which originally defines a slightly different MCS scheme for the PUSCH. The MCS can be quasi-statically selected at the UL transmitter and RX in the range between 0 and 28.

Note: Since no UL scheduling information is signaled from eNodeB to UE through the PDCCH, PUSCH resource allocation as well as PUSCH MCS must be manually configured at both devices.

2.2.3.2 DMRS

According to the LTE standard, DMRS are associated with the transmission of PUSCH or PUCCH. Since the application framework does not provide a PUCCH implementation, only DMRS associated with PUSCH are supported. For the supported frame structure and the supported normal CP mode, DMRS are transmitted in the fourth and the eleventh OFDM symbol of an UL subframe, but only in the PRBs that carry PUSCH (refer to Figure 9).

The DMRS sequence generation implemented in the application framework is slightly simplified in comparison to the specification in *TS36.211: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 10)* [1]. In deviation to the LTE standard, DMRS sequences of different lengths (required

for different numbers of allocated PUSCH PRBs) are all derived from one fixed base sequence defined for 100 PUSCH PRBs (that is, for the maximum supported number of PUSCH PRBs). Shorter DMRS sequences are derived by taking the maximum length base sequence and cutting surplus symbols at the end. The base sequence generation itself is compliant with sections 5.5.1 and 5.5.1.1 of *TS36.211: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 10)* [1]. The following fixed parameter set is applied for the base sequence generation:

$$M_{SC}^{RS} = mN_{SC}^{RB}, \alpha = 0, u = 0, v = 0;\ m = 100,\ N_{SC}^{RB} = 12, N_{ZC}^{RS} = 1193$$

Since the sequence-group number u as well as the base sequence number v are both fixed to 0, this implies that neither group hopping nor sequence hopping are supported.

2.2.3.3 SRS

In the application framework UL, SRS support can be globally enabled or disabled. When SRS support is enabled, the last OFDM symbol in every UL subframe will be reserved for SRS transmissions, so it cannot be used for any other UL signal or channels anymore. This statement is true even if the UL TX is not actively transmitting SRS in the corresponding UL subframes. In TDD mode, the last two symbols of a special subframe can be used for SRS transmissions.

Active SRS transmissions can be individually scheduled for every UL subframe and every special subframe per radio frame. According to section 5.5.3.3 of *TS36.211: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 10)* [1], this corresponds to the following:

- For FDD: *srs-SubframeConfig* = 0
- For TDD in general: *srs-SubframeConfig* = 7 (also equivalent to *srs-SubframeConfig* = 1 for TDD5/5)

For this purpose, a special parameter is provided at the control interfaces of the UL TX and UL RX. It is a bit vector with 10 elements, and each element addresses one specific subframe of a radio frame. In TDD mode, this bit vector will be masked with the supported pattern of special and UL subframes to ensure that active SRS transmissions are only possible in these subframes. When individually addressing the two SRS symbols in the UL pilot time slot (UpPTS) field of a special subframe, the following rule is applied in TDD mode:

- The bit related to a special subframe controls SRS transmissions in the last OFDM symbol of the special subframe.
- The bit related to the DL subframe preceding a special subframe controls SRS transmission in the previous last OFDM symbol of that special subframe.

The used SRS bandwidth is always fixed to 96 PRBs, which corresponds to SRS bandwidth configuration C_SRS = 0 and SRS bandwidth B_SRS = 0.

The SRS transmission comb k_TC can be directly configured to be 0 or 1.
SRS frequency hopping is not supported.

The SRS sequence generation is implemented in compliance with sections 5.5.3.1 and 5.5.1 of *TS36.211: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 10)* [1], but only the following fixed parameter set is supported:

$$M_{SC}^{RS} = mN_{SC}^{RB} = 576, \alpha = 0, u = 0, v = 0;\ m = 48,\ N_{SC}^{RB} = 12, N_{ZC}^{RS} = 571$$

As explained in section **Error! Reference source not found.** for the DMRS, this implies that neither group hopping nor sequence hopping are supported.

The SRS transmitter is fully implemented. On the receiver side the SRS subcarrier data are extracted, but currently no further receiver operation is implemented.

2.2.3.4 Supported UL Resource Grid

Figure 9 shows exemplarily the supported resource grid for an UL subframe with active SRS.

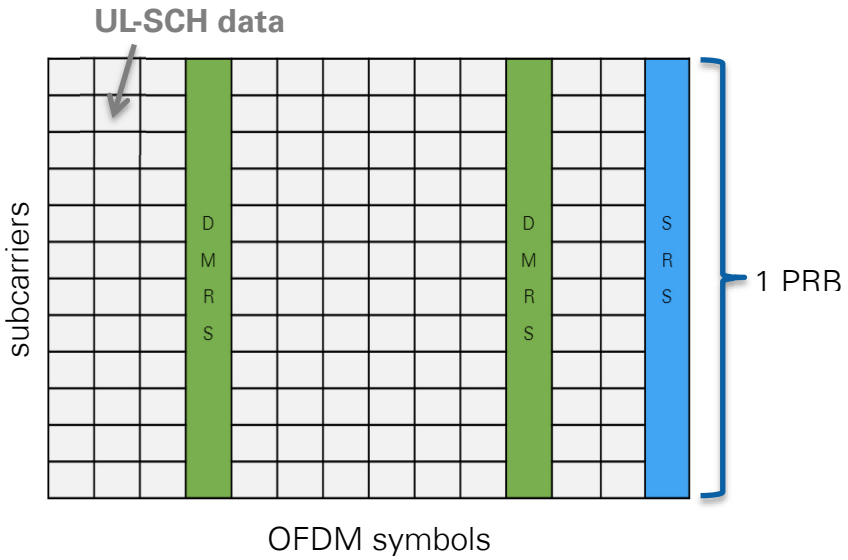

Figure 9: Time-Frequency Resource Grid of an UL Subframe with Enabled SRS Support

2.2.3.5 UL Transmit Timing Control

You can control the UL TX timing at the UE transmitter in a quasi-static manner using the UL timing advance parameter. In the application framework, the UL timing advance is set to zero by default. Zero means that the start of the transmitted UL radio frame is fully aligned to the start of the received DL radio frame at the UE antenna connectors. To cope with the propagation delay of real radio channels, the start of the UL can be advanced from 0 up to 30,719 baseband samples (that is, from 0…30,719xTs with Ts = 1/(30.72 MHz).

Note: *TS36.211: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 10)* [1] defines a fixed timing advance offset $N_{TAoffset}$ = 624xTs for TDD. This offset is not automatically applied by the application framework in TDD mode but can be manually set if needed.

In addition to the UL timing advance mechanism, the application framework autonomously corrects the UL TX timing in relation to the DL radio frame timing measured at the UE RX. Thus, time tracking steps applied in the DL RX are also applied to the UL TX. The time tracking is designed to cope with clock frequency differences between eNB and UE of up to ±5 ppm. Potential timing corrections are applied once per radio frame at the beginning of the radio frame.

They are realized by reducing or extending the cyclic prefix by an integer number of baseband samples. The current implementation allows for maximum correction steps of up to ±160 baseband samples (±160 x Ts). As long as the observed clock frequency difference is below the target maximum of ±5 ppm also, the maximally applied timing correction step should be below ±2 x Ts, which is the upper limit defined by the LTE standard.

2.2.3.6 UL Frequency Offset Correction

In addition to the autonomous UL TX timing correction, the application framework also applies an autonomous UL frequency offset correction. Carrier frequency offsets measured and corrected in the DL RX are also applied in the UL TX with a carrier frequency depending scaling factor: $1\times f_{C,UL}/f_{C,DL}$. Notice that frequency shifts in the UL TX must be applied in the opposite direction as in the DL RX. For FDD, the ratio between UL carrier frequency and DL carrier frequency must be considered.

2.3 MAC for Windows Host and FPGA

The application framework implements the necessary functionality for establishing a link between the eNB (DL TX) and UE (DL RX) and for enabling packet-based data transmission in the DL. Furthermore, it provides feedback of DL channel state information and DL (HARQ) ACK/NACK through the UL, as well as basic adaptive modulation and coding (AMC), known as link adaptation in DL, enabling DL closed loop operations

2.3.1 System Configuration

The system configuration is fixed to the following values at both the DL TX (eNodeB) and DL RX (UE):

- Bandwidth: 20 MHZ (100 PRBs)
- Cell-ID: 0
- Control Format: 1 (PDCCH spans over one OFDM symbol per subframe)
- Antenna configuration: single antenna (single-input, single-output (SISO))

Because of the fixed configuration, System Information Block (SIB) transmission and reception is not needed and therefore not implemented.

2.3.2 DL Scheduling

The application framework supports a quasi-static scheduling of the physical DL shared channel (PDSCH) with respect to the resource block allocation. The PDSCH resource allocation can be controlled through a special control at the eNB (DL) transmitter. It is valid when the DL TX is active. It can only be changed when the DL TX is deactivated. The PDSCH MCS can be controlled in two alternative ways:

- Quasi-statically by means of a special control at the DL TX (eNB)
- Automatically by means of the rate adaptation functionality of the DL MAC. This is based on the DL channel state feedback information (wideband signal-to-interference-plus-noise ratio (SINR)) received through the UL. The DL MCS will be adapted to obtain a default target block error rate (BLER) at the DL RX of about 5-10%. A special parameter "SINR offset [dB]" can be used to indirectly control the target DL BLER by reducing the reported SINR by the given value before it is fed into the rate adaptation framework.

The PDCCH is used for signaling the PDSCH configuration (resource block allocation and MCS) from the eNB (DL) transmitter to the UE (DL) RX. The signaling of the so-called DCI is done for every DL subframe and every special subframe. Since PDSCH transmissions in special subframes are not currently supported by the application framework, no DCI message is transferred in those subframes.

Signaling the DCI through the PDCCH allows the UE (DL) RX to be automatically configured to the PDSCH transmission parameters that might be dynamically selected by the eNB MAC. This is a pre-requisite for dynamic link adaptation (ACM) in the DL.

In principle, LTE eNBs are designed to support multiple UEs. That's why the CRC attached to the DCI messages during the encoding is masked with a UE specific radio network temporary identifier (RNTI). During the decoding of the PDCCH the UE receiver checks whether this CRC mask fits with its own RNTI or not. If mismatched, the DCI message will be discarded and the PDSCH data will not be decoded. This way the eNB can address a specific UE so that only this device decodes the PDSCH data. This also implies that for a successful DL data transmission with the application framework, the RNTI selected at the eNB transmitter has to be identical to the RNTI set at the UE receiver.

2.3.3 UL Scheduling

Similar to the DL, the UL uses a quasi-static scheduling. For the PUSCH, both the resource block allocation and the MCS must be configured manually. Since the PDCCH does not support signaling of UL scheduling information, both the eNB receiver as well as the UE transmitter must be configured manually.

As described previously, the eNB DL TX starts transmitting (with a quasi-statically selected configuration) as soon as it is switched on. Assuming the UE DL RX is configured accordingly (that is, same carrier frequency, same frame structure, same reference symbol type, and so on), it should be able to synchronize to the DL TX and to receive and decode the DL PHY channels. The same holds for the UL. The UE transmits a quasi-statically configured UL as soon as it switched on, and the eNB receiver should be able to receive and to decode the UL transmission as long as it is configured correctly. All adjustable parameters like carrier frequency, frame structure, transmit power, and UL timing advance must be configured manually. Complex cell search or cell attachment procedures are not implemented.

2.3.4 DL MAC Packet for User-Defined Data

A simple MAC implementation (“Mini MAC”) is used to fill the transport blocks of the DL shared channel with user-defined payload data. This way, the DL can be used for packet-based data transmission.

The MAC packet format is proprietary and shown in Figure 10. The PDSCH transport block size is defined according to tables 7.1.7.1-1 and 7.1.7.2.1-1 of *TS36.213: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (Release 10), V10.12.0* [3]. It depends on the used modulation and coding scheme (MCS) and the number of resource blocks allocated for the PDSCH.

For each subframe the simple MAC implementation checks the fill state of the payload data FIFO that is filled with user-defined data from the host. Depending on the buffer fill state and the number of PDSCH transport block bits usable in the current subframe, the MAC determines the number of payload bytes that can be effectively transmitted by the PDSCH transport block. Based on this, the content of the actual PDSCH transport block is composed. It starts with a 4 byte long MAC header that contains the number of effectively transmitted payload bytes. The second part is the payload itself. If necessary, zero padding bits are added to fill up the PDSCH transport block to the configured size.

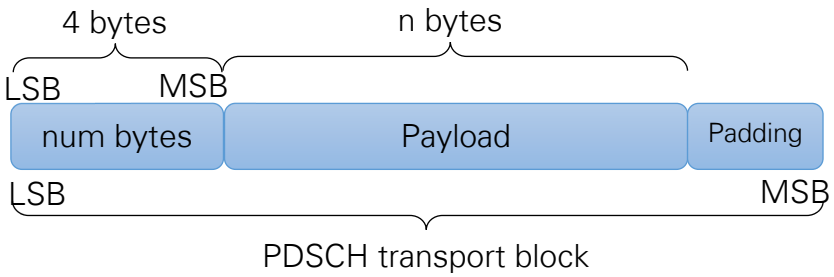

Figure 10: MAC Packet Structure

2.3.5 UL MAC Packet for Feedback Information

The content of the UL transport blocks is composed based on the same MAC packet format as described for the DL (refer to Figure 10 above). The actual *payload* portion is filled with DL feedback information. As shown in Figure 19, each *payload* portion of an UL MAC packet comprises 6 words of 32 bits, that is, 192 bits in total. Unused bits are set to zero.

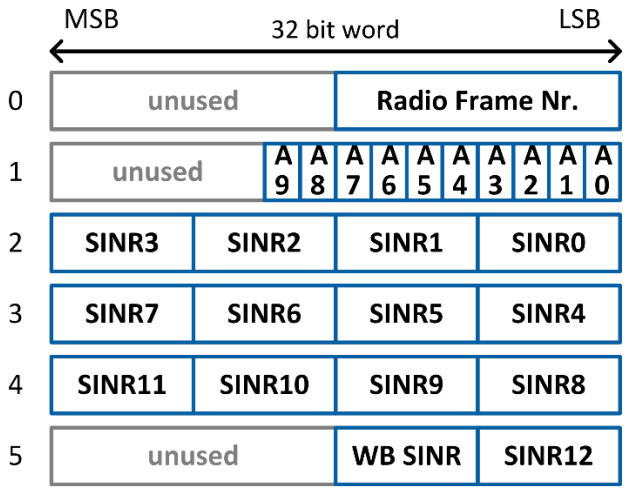

Figure 11: Feedback Information Packet Format

The feedback information is generated by the UE host implementation (UE feedback MAC) based on results provided by the UE DL RX. The following feedback information is provided:

- Radio Frame Number: A counter value from 0 to 1,023 that is increased for each radio frame. It corresponds to the system frame number (SFN) defined in the LTE standard with the following difference: the timing is generated on the UE side and not dictated from the eNodeB using system information.
- A0 – A9: the DL ACK/NACK/DTX information for the latest 10 subframes providing information about the success of the PDSCH reception.
  The values are encoded as a 2 bit number:
  - 0 = NACK (PDSCH was received with CRC error)
  - 1 = ACK (PDSCH was received with CRC ok)
  - 2 = DTX (no PDSCH was decoded because of missing or invalid DCI message)
  - 3 = undefined
- Subband and wideband SINR in dB
  - Subband size used for SINR calculation: 8 PRBs
  - Subband numbering:
    - SINR0 is subband SINR for PRBs 0..7, SINR1for PRBs 8..15, and so on
    - SINR WB is the wide band SINR calculated over all subbands
  - Fixed point format: 8 bits signed fixed-point number with 6 integer and 2 fractional bits (range -32.00 dB to +31.75 dB)
  - SINR calculation is based on the CRS- or UERS-based channel estimates provided by the DL RX. For further details, refer to section 2.3.5.1. When using UERS the resource block allocation is not currently considered for SINR calculation. As UERS are only transmitted on allocated PRBs, SINR calculation results on not or only partially allocated subbands are undefined (that is in most

cases less than the actual channel quality). Wideband SINR value is also undefined if not all subbands are fully occupied.

2.3.5.1 SINR Estimation Algorithm

The SINR estimation algorithm is based on filtering the potentially noise channel estimates derived for the CRS subcarrier or the UERS subcarrier, respectively.

The noisy least squares (LS) channel estimates obtained for the reference symbol carriers are filtered by a de-noising low-pass filter to obtain LS channel estimates with reduced noise. The implemented prototype de-noising filter is a raised cosine filter with nine taps and the filter coefficients in the following table:

Coefficient Index	Coefficient value
0	0
1	-0,061235467
2	0
3	0,306177333
4	0,510116268
5	0,306177333
6	0
7	-0,061235467
8	0

The average signal power of the difference between the noisy channel estimates and their low-pass filter complements can be taken as raw estimate for the noise variance; that is, it can be considered as scaled version of the actual noise variance in the given frequency band or sub-band.

Averaging the squared magnitudes of the low-pass filtered channel estimates delivers a raw estimate of the mean channel power gain.

Based on the known reference signal transmit power, the raw estimate of the mean channel power gain, and the raw estimate of the noise variance, a raw carrier to interference noise ratio (CINR) estimate can be derived.

Because both the raw mean channel power gain estimates and the raw noise variance estimates are biased, the raw SINR estimates are biased. Notice that especially the bias of the noise variance estimates strongly depends on the actual noise variance. That's why a mapping function must be applied to map the biased raw SINR estimates to the final, so-called unbiased SINR estimates.

This mapping function has been derived by means of calibration simulations and measurements. It has been approximated by the set of the following two linear functions, which are valid for a target SINR range between -6...30dB):

$$SINR/dB = 1.8*SINRraw/dB - 10.2 \quad \text{for } -6 \geq SINRraw/dB < 6$$

$$SINR/dB = 1.1*SINRraw/dB - 6 \quad \text{for } 6 \leq SINRraw/dB \leq 30$$

Note: To further improve the provided SINR estimates an additional look-up table based fine-calibration stage is implemented in the application framework. The underlying look-up table has been derived by fine-calibration measurements.

2.4 Interfacing PHY and MAC on Real-Time Targets

On real-time (RT) targets such as the USRP-2974 Software Defined Radio Stand-Alone Device, the application framework offers an open API to connect arbitrary MAC implementations with the framework PHY. The API adds an PHY service access point (SAP) on top of the PHY. The SAP is a message interface based on byte streams. The messages are explained in more detail in section 5.1.

2.4.1 PHY SAP

The PHY SAP enables easy adaptation of different MAC implementations. The PHY SAP provides a message-based interface between the LTE L2 and L1 stack. It includes the eNodeB and UE with DL and UL transmission.

The PHY SAP follows a general API concept with the following principles:

- The communication is based on a common set of 3 general message types
  - REQ: Service request from higher layer to lower layer
  - CNF: Confirmation of the received service request from lower layer to higher layer
    - With the option to be disabled
    - Confirmation or REQ messages
  - IND: indication (status, data, error, and so on) from lower layer to higher layer
- For requests, control and payload are separated into separate messages
- Hierarchical message approach is used (section 2.4.1.1)

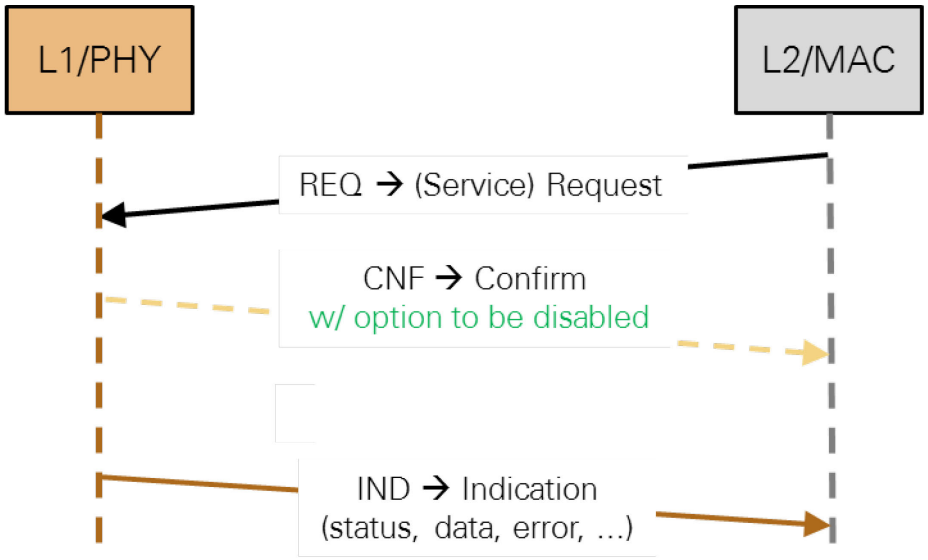

Figure 12 Simplified Communication between L1 and L2 Based on Two Basic API Message Types

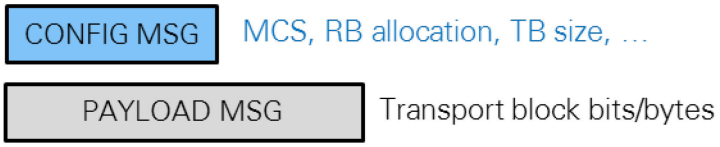

Figure 13 Requests Separated into CONFIG and PAYLOAD Messages

2.4.1.1 PHY SAP Message Definitions

In this section all PHY SAP messages required to control the application framework L1 are specified. The defined sequence of messages and the corresponding state machine is provided in sections 4 and 5.

2.4.1.1.1 PHY SAP General Message Structure

The message definitions follow a generalized structure use in the application frameworks (Figure 14), which follow these design rules:

- All fields in an API message are byte aligned, that is, they consist of one or multiple bytes.
- Every API message contains an integer number of bytes.
- If the physical transport mechanism of specific API implementation requires the API messages to be aligned to words of multiple bytes (for example, U32, U64), the physical transport layer of the API is responsible for adding and removing padding bytes as needed. These padding bytes are not part of the logical API message definition, that is, they will not contribute to the message body length signaled through the related field in the general message header.

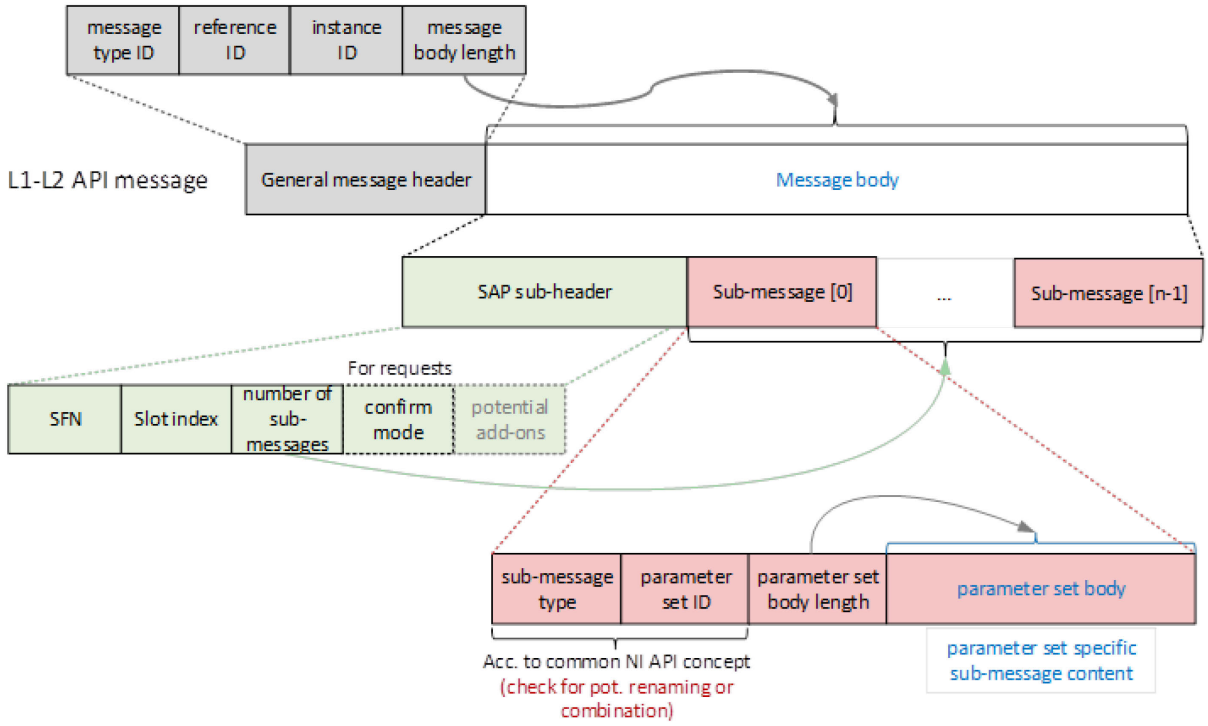

Figure 14 General Message Structure

For details about the messages defined in the current version of the application framework, refer to section 5.1.5.1.

When transferring messages as a stream, the API uses network byte order (big endian, highest byte first) to be independent from hardware platform. Bit order within bytes is MSB first because this is the natural bit order in connection with network byte order, as shown in the figure below.

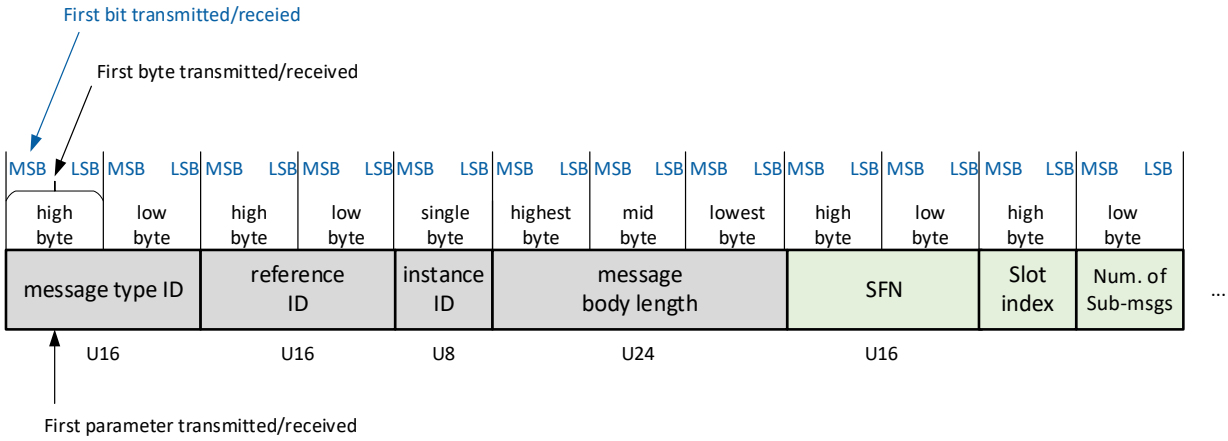

Figure 15 Illustration of Byte Order and Bit Order Defined for the Transmission of Messages

2.4.2 Confirmation Handling

Although the API allows you to switch off confirmation message handling, confirmation messages are mandatory in the current implementation.

2.4.2.1 Logical State Machine for Common Request and Confirmation

This logical state machine defines the logical sequence of steps for the reception, validation, and application of a common single PHY Request message as well as the generation of the related PHY_CNF messages.

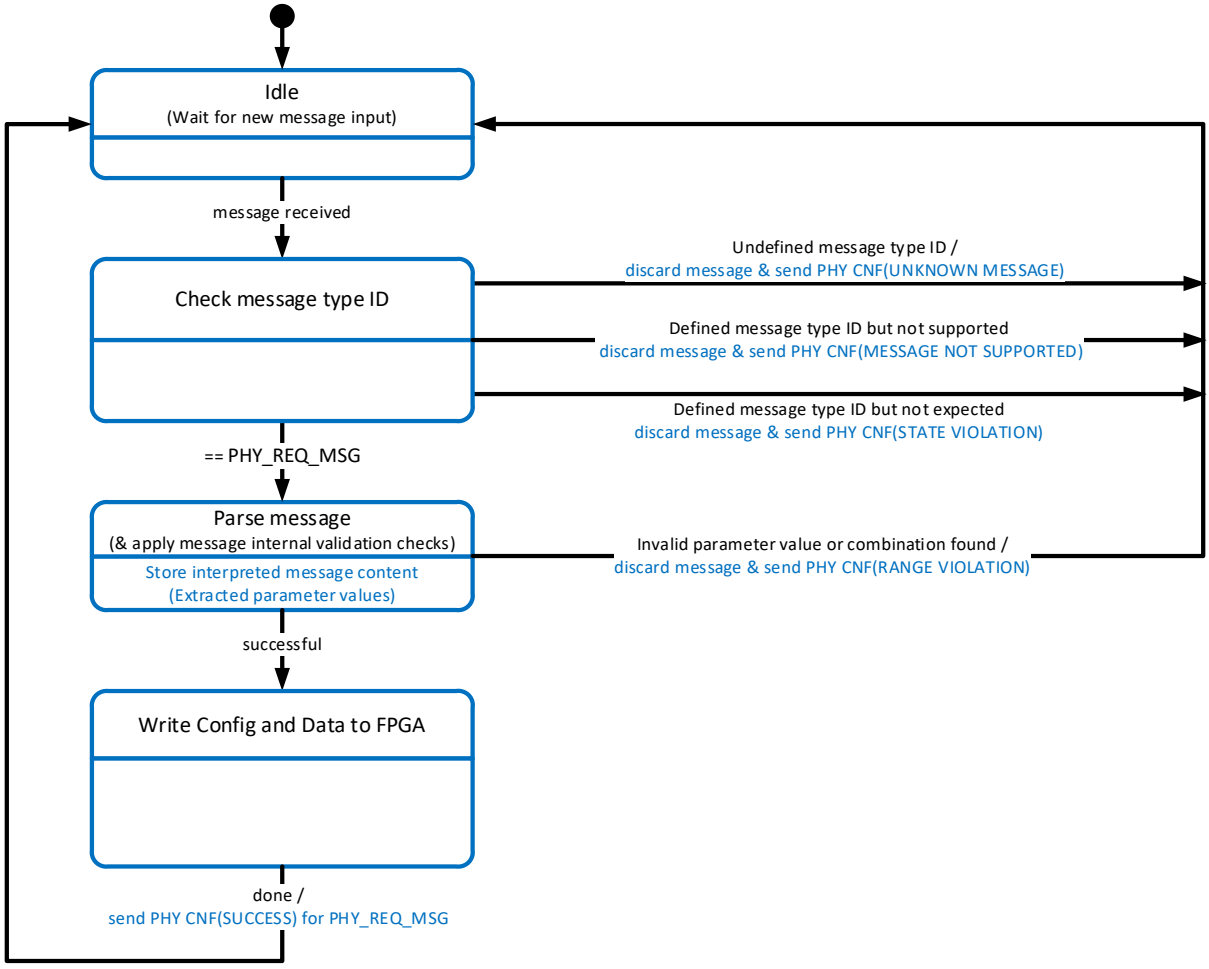

Figure 16 Logical State Machine for Common PHY Request and Confirmation Handling

2.4.2.2 Logical State Machine for Downlink TX Request and Confirmation

This logical state machine defines the logical sequence of steps for the reception, validation, and application of a DL TX Request message sequence at the eNodeB (PHY_DL_TX_CONFIG_REQ and PHY_DL_TX_PAYLOAD_REQ) as well as for the generation of the related PHY_CNF messages. It further defines the conditions for generating a specific confirmation status. A corresponding DL TX Request message sequence is shown in section 4. The same procedure can also be applied for an UL Tx Request message sequence at the UE (PHY_UL_TX_CONFIG_REQ and PHY_UL_TX_PAYLOAD_REQ). The current implementation does not support PHY_UL_TX_CONFIG_REQ, hence the UL configuration is static.

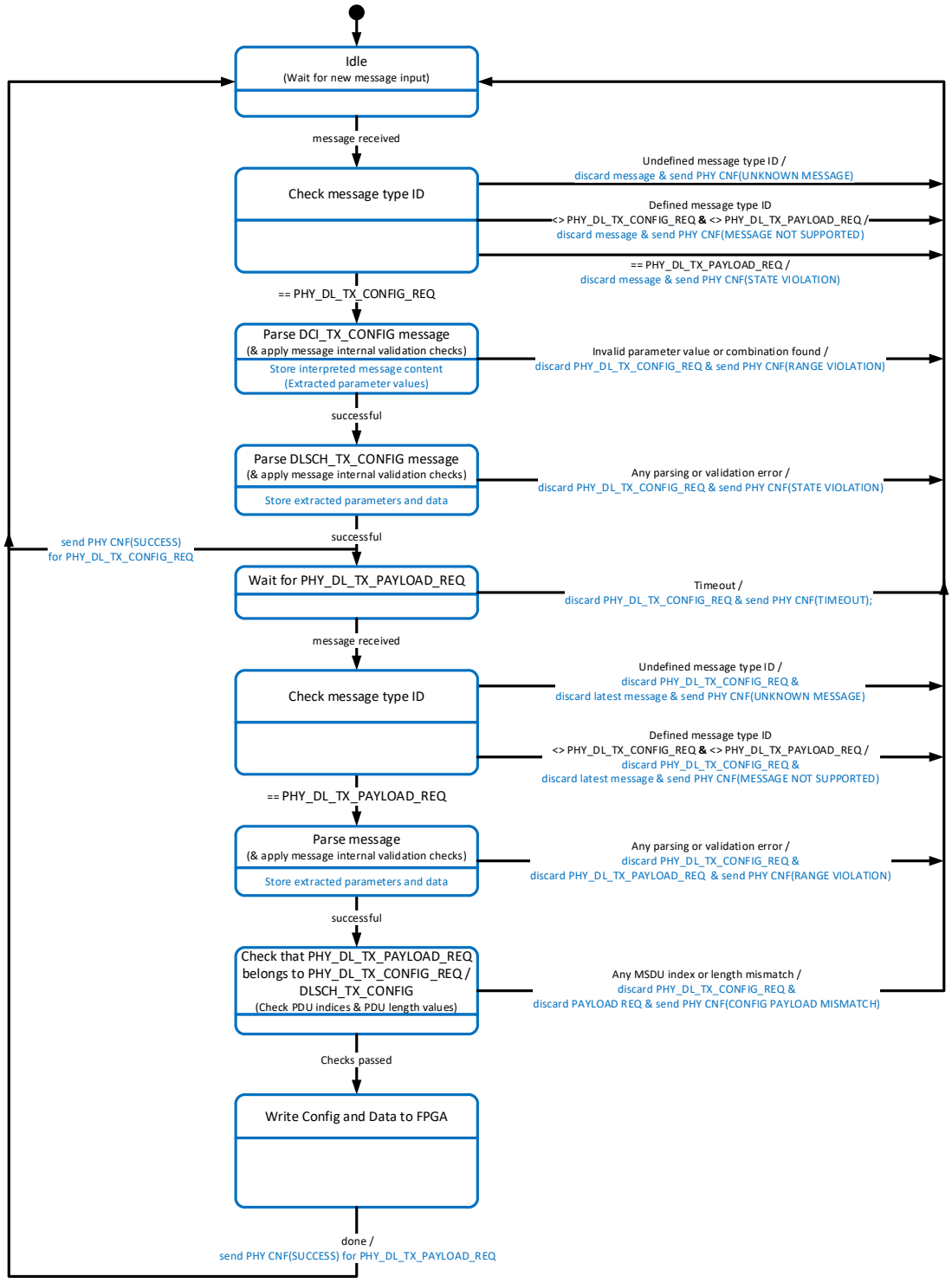

Figure 17 Logical State Machine for PHY TX Request and Confirmation Handling

2.4.2.3 General Confirmation Message Handling

The following subsections list validation checks that must be applied for every incoming PHY TX request, that is, every incoming message pair of PHY_DL_TX_CONFIG_REQ and PHY_DL_TX_PAYLOAD_REQ.

Whenever a validation check fails the following actions must be applied:

1. The related message (or message pair) is discarded/rejected.

2. PHY CNF message is generated per discarded message with the following related negative confirmation status: Check of correct message sequence

Done according to the state machine.

Common message validation checks are in the following table:

	Validation condition (pass criteria)	**(Negative) Confirmation status if check not passed**
Checks applied on general message header, SAP sub-header, overall message		
	Received message type ID is defined (section 3.3)	UNKNOWN MESSAGE
	Received message type is supported, that is, that belongs to the set of a certain message sequence	MESSAGE NOT SUPPORTED
	<number of sub-message> is in supported value range (section 3)	RANGE VIOLATION
	<confirm mode> is in allowed range	RANGE VIOLATION
	A specific sub-message (parameter set) is contained in the message only once (not multiple times)	PARAMETER (SET) REPETITION
For each sub-message (parameter set) contained in the received message		
	received <sub-message type> and <parameter set ID) values are defined for the specific message	UNKNOWN PARAMETER (SET)
	all parameters defined for the sub-message (parameter set) are contained in (that is, can be extracted from) the received sub-message (parameter set) **Note:** This is also some kind of indirect length validation check.	MISSING PARAMETER (SET)
For each parameter per sub-message (parameter set)		
	parameter value is in in the allowed range (section 3)	RANGE VIOLATION

The CONFIG vs. PAYLOAD message consistency check is described in the following table.

Validation condition (pass criteria)	**(Negative) Confirmation status if check not passed**
Checks applied on every received CONFIG, PAYLOAD message pair	

Validation condition (pass criteria)		(Negative) Confirmation status if check not passed
	All PDU indices extracted from the parameter sets of the CONFIG message must be equal to the PDU index extracted from the PAYLOAD message	CONFIG PAYLOAD MISMATCH
	The PDU length extracted from the CONFIG message has to match to the PDU length extracted from the payload message	CONFIG PAYLOAD MISMATCH

3 Implementation Details

3.1 Architecture

Figure 18 and Figure 19 show the block diagram of the system in the DL, eNodeB and UE operation modes. Data streams that require high data rates for data transfer between host and FPGA are implemented as DMA FIFOs. These streams include the payload and UL data from host to FPGA and the received PDSCH/PUSCH transport blocks from FPGA to host. In-phase/quadrature (I/Q) samples for constellation and spectrum display as well as the channel estimation values are also transferred from FPGA to host using DMA FIFOs. Further status information is transferred to the host by reading the indicator values.

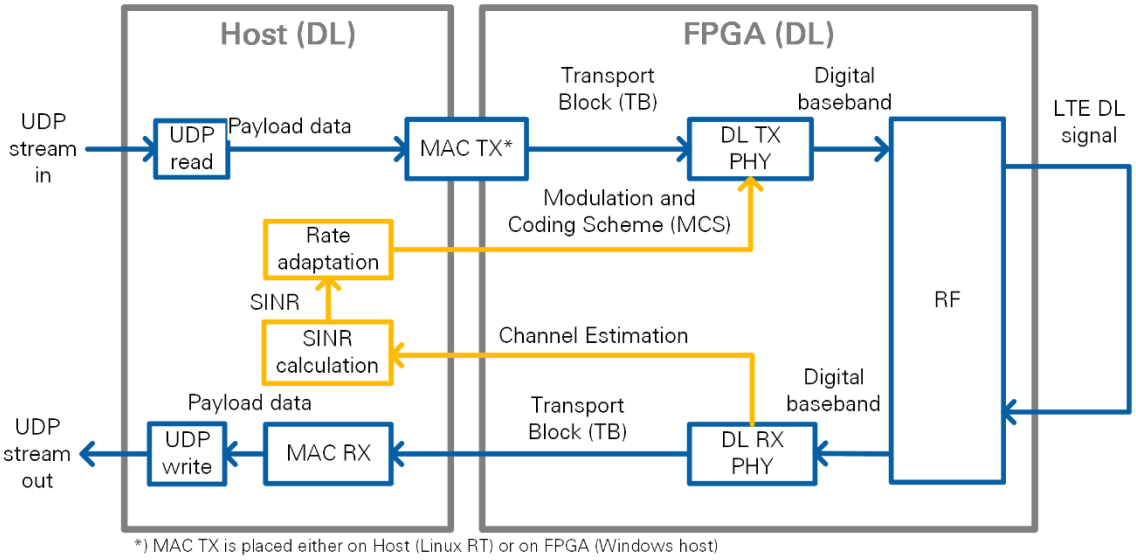

Figure 18: Block Diagram of the System in DL Operation Mode (Single-Device Setup)

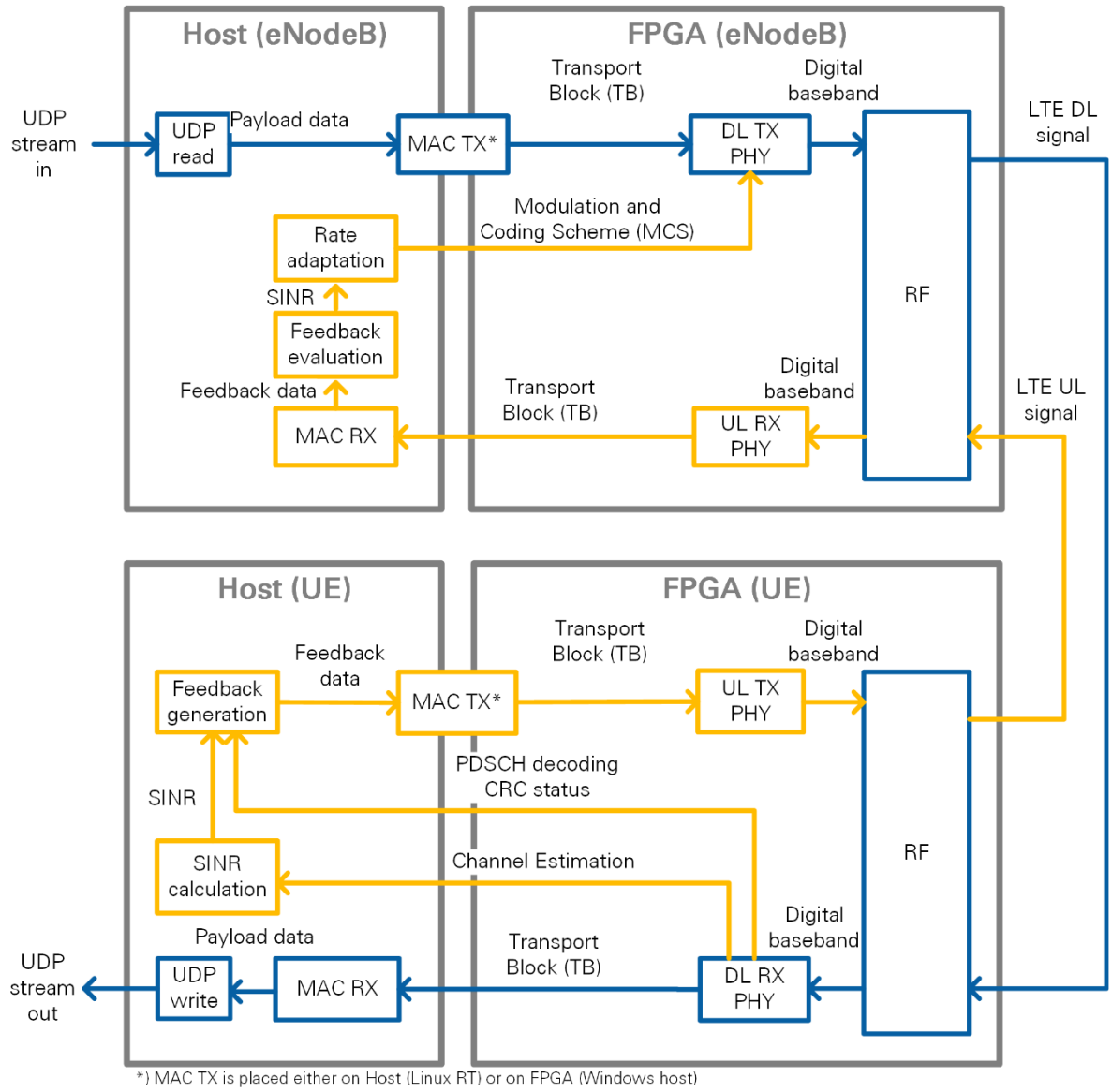

Figure 19: Block Diagram of the System in eNodeB/UE Operation Mode (Double-Device Setup)

The components shown in Figure 18 and Figure 19 perform the following tasks:

- **UDP read:** Reads data, provided by an external application, from a user datagram protocol (UDP) socket. The data is used as payload data in the transport block (TB), which is then encoded and modulated as an LTE DL signal by the DL TX PHY.
- **UDP write:** Writes the payload data, which was received and decoded from the LTE DL signal by the DL RX PHY, to a UDP socket. The data can then be read by an external application.
- **MAC TX:** On FPGA, a simple MAC implementation that adds a header to the TB containing the number of payload bytes. The header is followed by the payload bytes, and the remaining bits of the TB are filled with padding bits. Using MAC on an RT payload is assembled by packet. The TB needs remaining bytes of two bytes length information and packet length.
- **MAC RX:** Disassembles the TB and extracts the payload bytes.

- **DL TX PHY:** PHY of the DL TX. Encodes the physical channels and creates the LTE DL signal as digital baseband I/Q data. This includes encoding of the PDCCH, encoding of the PDSCH, resource mapping, OFDM modulation.
- **DL RX PHY:** PHY of the DL RX. Demodulates the LTE DL signal and decodes the physical channels. This includes primary synchronization sequence (PSS) based synchronization, OFDM demodulation, resource demapping, channel estimation and equalization, decoding of the PDCCH, and decoding of the PDSCH.
- **UL TX PHY:** PHY of the UL TX. Encodes the physical channels and creates the LTE UL signal as digital baseband I/Q data. This includes encoding of the PUSCH, resource mapping, and OFDM modulation.
- **UL RX PHY:** PHY of the UL RX. Demodulates the LTE DL signal and decodes the physical channels. This includes OFDM demodulation, resource demapping, channel estimation & equalization, and decoding of the PUSCH.
- **SINR calculation:** Calculation of the SINR based on the channel estimation which was used for PDSCH decoding. Channel estimation is based on either CRS or on UERS.
- **Rate adaptation:** Sets the MCS depending on the measured/reported SINR. The aim is to ensure that the BLER of the PDSCH decoding is kept low.
- **Feedback generation:** Creates a feedback message that contains the measured subband and wideband SINR, as well as the ACK/NACK information (the CRC result of the PDSCH decoding) of the previously received radio frame.
- **Feedback evaluation:** Extracts the subband and wideband SINR, as well as the ACK/NACK information from the feedback message.

3.2 FPGA Implementation

This application framework is based on the target specific sample streaming project (*PXIe USRP RIO 40 MHz BW Single-Device Streaming* for USRP devices with 40 MHz bandwidth, *PXIe USRP RIO 120 MHz BW Single-Device Streaming* for USRP devices with 120 MHz bandwidth, or *PXIe NI-579xR Streaming* for FlexRIO/FlexRIO adapter modules (FAM)). The sample streaming project contains the basic logic to interface with the analog-to-digital converter (ADC) and digital-to-analog converter (DAC) registers. It also performs digital up and down conversion, configuration for the front-ends, and RF impairment correction.

In the application framework, the processing blocks for the DL and UL TX and RX are implemented on the FPGA directly. They exchange the baseband data with the RF interface using target-scoped FIFOs. The processing on the FPGA has advantages because it provides lower latency and therefore enables real-time physical layer processing. This approach is different from the sample streaming project where the digital baseband data is sent to or received from the host, which is then responsible for all channel encoding and decoding.

Figure 20 shows the structure of the FPGA implementation for the different operation modes. The outer boxes (highlighted in light gray) represent single-cycle

timed loops that implement clock-driven logic. The inner boxes correspond to the high-level blocks described in the architectural overview. The transmitter loop receives payload data from the host through a DMA FIFO, performs channel encoding, and generates the TX baseband signal, which is passed to the RF loop for upconversion. The RF loop is inherited from the sample streaming project. It also performs downconversion of the RX baseband signal that is passed to the receiver loop for channel decoding. The decoded transport blocks are sent to the host using a DMA FIFO.

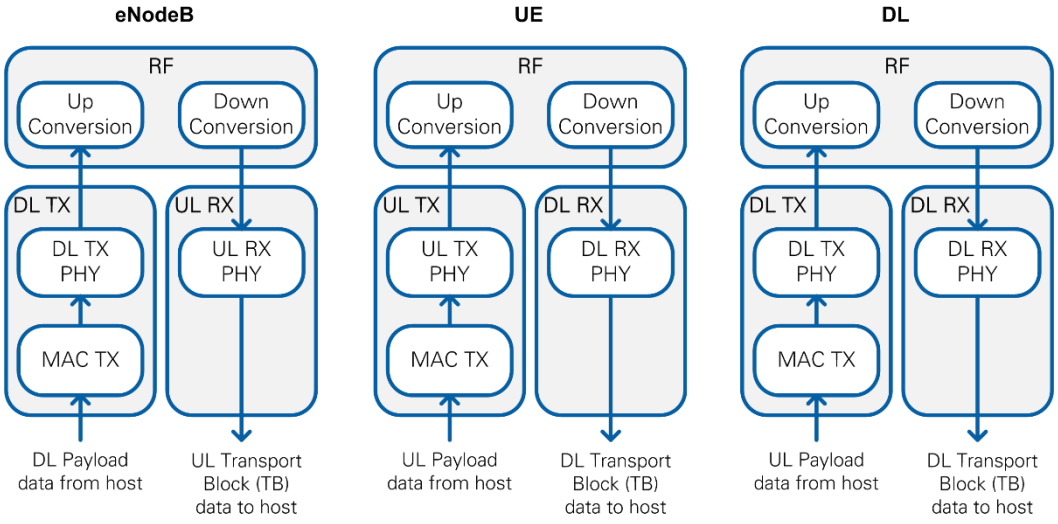

Figure 20: Implemented FPGA Loops for Different Operation Modes

3.2.1 RF Interface

3.2.1.1 Digital Upconversion and RF Impairment Correction

The transmit part of the RF loop is shown in Figure 21. The first block is the fractional interpolator, which converts the standard LTE rate of 30.72 MS/s to the DAC sample rate (120 MHz for USRP RIO 40 MHz BW, 200 MHz for all other USRP devices, and 130 MHz for FlexRIO/FAM). Next, the frequency shift module performs a fine frequency shift, which is automatically configured by the RF driver. The I/Q imbalance correction uses coefficients determined during manufacturing and stored in the device EEPROM.

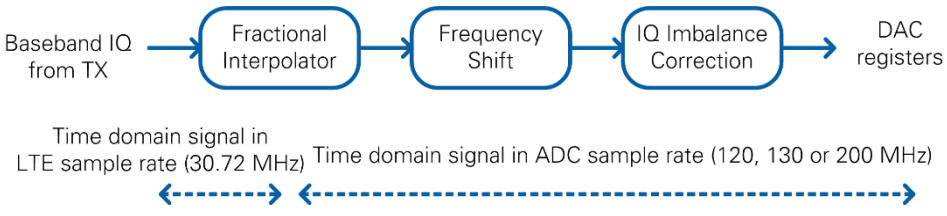

Figure 21: Digital Upconversion Block Diagram

3.2.1.2 Digital Downconversion and RF Impairment Correction

The receive part of the RF loop is shown in Figure 22. It corrects I/Q impairments in the baseband signal, performs fine-frequency adjustments, and performs sample rate conversion. I/Q imbalance correction uses coefficients determined during

manufacturing and stored in the device EEPROM. A decimator converts the sample rate from the ADC sample rate (120 MHz for USRP RIO 40 MHz BW, 200 MHz for all other USRP devices, 130 MHz for FlexRIO/FAM) to the standard LTE rate of 30.72 MS/s. The application framework also includes a DC suppression to compensate for the DC offset. This module averages over time to remove the DC portion of the signal.

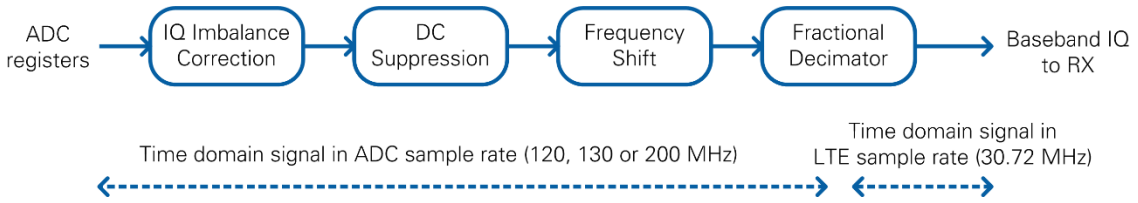

Figure 22: Digital Downconversion Block Diagram

3.2.2 DL TX

The DL TX is implemented by the FPGA top-level variants eNodeB and DL. In the block diagrams of Figure 18 and Figure 19, it corresponds to the DL TX PHY block.

As shown in the simplified block diagram in**Error! Reference source not found.**, it performs the following tasks:

- PDCCH encoding
- PDSCH encoding
- Mapping to resource elements
- IFFT conversion + CP conversion (OFDM modulation)

The following reference signals and physical channels are mapped:

- PSS
- CRS
- UERS, if enabled
- PDCCH
- PDSCH

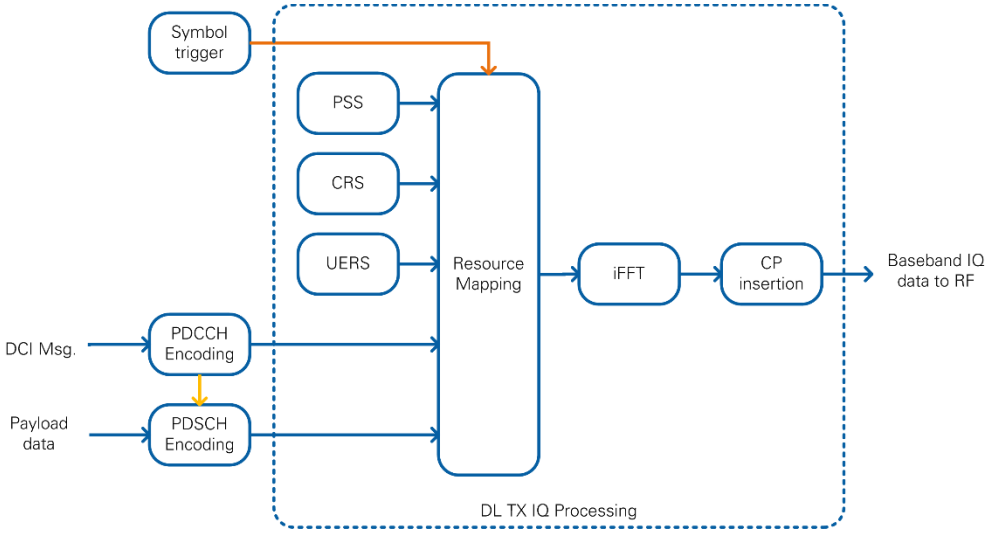

Figure 23: Simplified Block Diagram of the DL TX

Channel Encoding is performed for the PDCCH and PDSCH physical channels**Error! Reference source not found.**. The encoding is performed for each DL subframe by the FPGA implementation which allows real-time operation. The other signals are read out from look-up tables (LUTs).

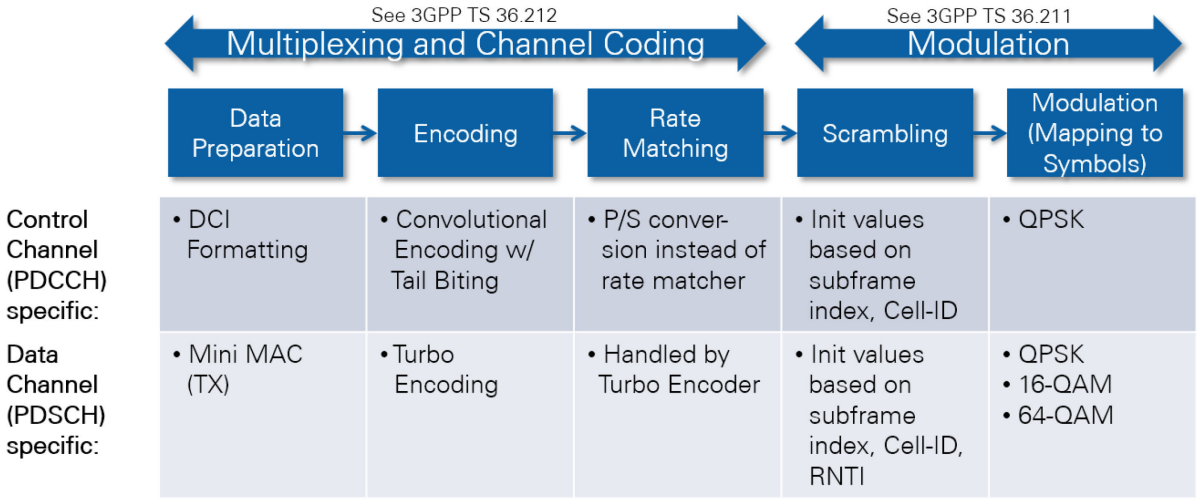

Figure 24: Channel Encoding Performed in the DL TX

3.2.2.1 FPGA Implementation Details

An extended block diagram that represents the actual implementation is shown in Figure 25. It shows the data path (blue), the configuration signals (yellow) and the triggering signals (orange).

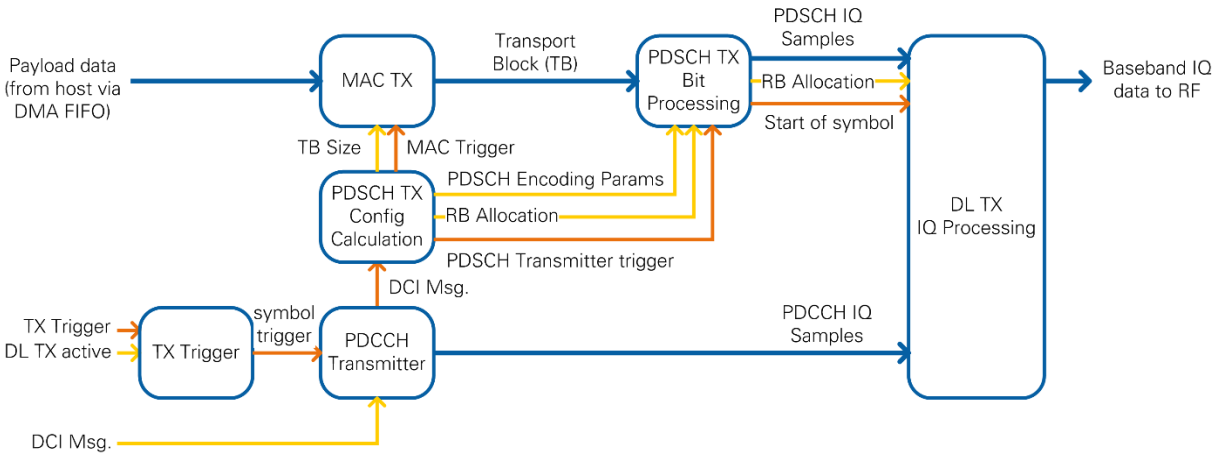

Figure 25: Block Diagram of the DL TX Showing Data, Trigger and Control Paths

If the DL TX is active and if it receives a TX trigger signal, it generates symbol triggers for one complete radio frame (10 subframes × 14 symbols/subframe = 140 symbols). The TX trigger signal is provided from a register that is written in the DAC/ADC loop. This ensures synchronization between both loops and compensates eventual clock drifts. The symbol trigger is passed through the rest of the chain and causes the modules to produce data. A subframe trigger is derived for the PDCCH and PDSCH transmitters that produce enough data for one subframe. The DL TX I/Q

uses the symbol trigger and generates time-domain I/Q samples for one OFDM symbol.

The PDCCH transmitter performs channel encoding for the PDCCH. The DCI message serves as input data. As shown in Figure 3, it contains the MCS and the RB allocation. The resulting PDCCH I/Q samples are written to a FIFO inside the TX IQ Processing module.

The DCI message is also input to the module which calculates the PDSCH transmitter configuration parameters (PDSCH TX Config Calculation). It performs two tasks. First, it calculates the TB size and triggers the MAC TX. Second, it calculates the PDSCH encoding parameters, extracts the RB allocation, and triggers the PDSCH TX Bit Processing module.

The MAC TX assembles the transport block in the format as shown in Figure 10. The payload data is read from the host through a host-to-target DMA FIFO.

The PDSCH TX bit processing module includes the channel encoding, scrambling and modulation of the physical DL shared channel (PDSCH). It uses the MCS defined inside the DCI message. The resulting PDSCH I/Q samples are written to a FIFO inside the TX IQ Processing module.

The TX IQ Processing module is triggered after the PDCCH and PDSCH I/Q samples for the current subframe are generated. It includes the resource mapping that assembles all 1,200 subcarriers of the current symbol. The index generator generates the timing information for each sample of the current OFDM symbol, such as subcarrier, resource block, OFDM symbol and subframe index. Depending on the current timing information, the index-to-channel mapping decides for each subcarrier, which reference signal or physical channel is mapped to it. The PSS sync sequence, the CRS and the UERS are precalculated and stored in an LUT. The PDCCH and PDSCH I/Q samples are read from a FIFO that was filled with all I/Q samples for the current subframe before the TX IQ Processing module was triggered. After combining all channels, the DC gap is inserted, and whitespace is added so that the resulting number of samples equals the FFT size of 2,048. The IFFT converts the frequency domain data into the time domain. Finally, the cyclic prefix is attached to the output of the IFFT. The resulting time-domain signal is transferred to the RF loop using a FIFO.

3.2.2.1.1 PDCCH Transmitter

The PDCCH Transmitter creates all QAM symbols for the PDCCH channel. The contained blocks are illustrated in Figure 26.

Upon reception of the symbol trigger, the DCI message is generated based on the configuration from the host. The message is encoded, and a CRC is attached. The DCI multiplexer module shifts the generated symbols to the correct position within the PDCCH channel. Afterwards, the whole channel data is scrambled and interleaved. After applying QPSK modulation to the symbols, the PDCCH I/Q-

samples are fed into a FIFO that stores them until they are picked up by the resource grid generation.

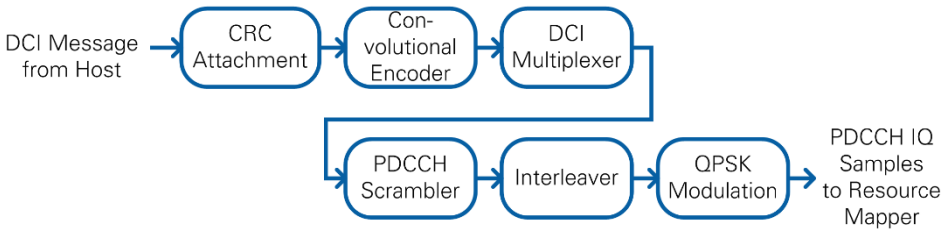

Figure 26: PDCCH Transmitter Block Diagram

3.2.2.1.2 PDSCH Transmitter

The PDSCH transmitter converts the user data from the host to QAM symbols for the PDSCH channel. The blocks of this processing chain are shown in Figure 27.

A basic MAC implementation adds a proprietary MAC header to the beginning of each PDSCH transport block. The remaining transport block is filled with as much user data as available from the host FIFO. If necessary, zero padding bits are added to fill the remaining portion of the transport block. The PDSCH Encoder uses the LTE DL Channel Encoder from Xilinx. The output of the core is scrambled according to the LTE specification. Afterwards, the QAM modulation is applied. The MCS value on the host sets the modulation scheme. After the modulation, the PDSCH QAM symbols are fed into a FIFO that holds these values until they are pulled into the resource grid.

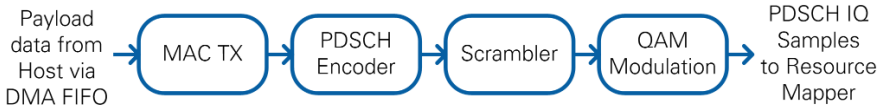

Figure 27: PDSCH Transmitter Block Diagram

3.2.3 DL RX

The DL RX is implemented by the FPGA top-level variants UE and DL. As shown in the simplified block diagram**Error! Reference source not found.**, it performs the following tasks:

- Synchronization and carrier frequency-offset (CFO) compensation
- CP removal + FFT conversion (OFDM demodulation)
- Demapping of the resource elements to the different physical channels
- CRS-based channel estimation and equalization
- UERS-based channel estimation and equalization
- PDCCH decoding
- PDSCH decoding

The following reference signals and physical channels are demapped:

- PSS
- CRS

- UERS
- PDCCH
- PDSCH

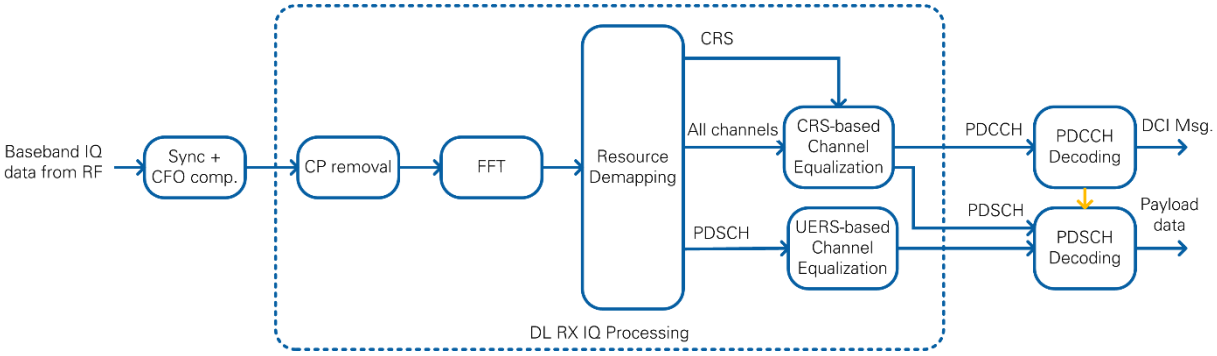

Figure 28: Simplified Block Diagram of the DL RX

The DL RX receives the I/Q samples in time domain from the RF loop that is derived from the sample streaming project. It performs the I/Q impairments correction, digital down conversion, frequency shift, and the down conversion from the ADC sample rate to the LTE sampling frequency of 30.72 MHz.

The first processing step in the DL RX loop is the synchronization. The PSS is used for radio-frame synchronization and CFO compensation.

The CRS are used for channel estimation and equalization. Per default, the CRS equalized samples are used for PDCCH and PDSCH decoding. The UERS can be used optionally. The UERS based channel estimation and equalization runs in parallel to the CRS based channel estimation and equalization. If UERS are enabled, the UERS equalized samples are used for PDSCH decoding.

The PDCCH decoder decodes the PDCCH. It includes the DCI message.

The PDSCH decoder decodes the PDSCH. The PDSCH configuration parameters are derived from the received DCI message. It determines the resource block allocation, that is, which subcarriers are filled with PDSCH, and the MCS. The decoded PDSCH transport blocks are written to the host using a target-to-host DMA FIFO.

The following sections describe each of the blocks in more detail.

3.2.3.1 Radio Frame Synchronization

Data is transferred from the RF loop to the DL RX loop using a FIFO. This block's main purpose is to determine the start of the LTE radio frame and to align the received time-domain signal accordingly. Another task of this block is the compensation of carrier-frequency offset. Only if both the sync signal was detected and the CFO was compensated, full radio frames are passed to the subsequent processing blocks.

The components of the Radio Frame Synchronization block are shown in Figure 29. Synchronization and CFO compensation are achieved by continuous measurement of both an autocorrelation and a cross correlation. LTE signals contain a PSS, which is detected by two finite impulse response (FIR) filters (real and imaginary part) that calculate the cross correlation. This operation is executed on a reduced sample rate of 1.92 MS/s, which is the result of a decimation by 16. For each radio frame, the cross correlation peak is detected. To avoid misdetection, a validation unit checks that the peak amplitude is eight times higher than the average energy of the cross correlation. Additionally, three consecutive peaks are required, and the peak position may not drift more than five samples.

In parallel to the cross correlation, an autocorrelation is performed on the full sample rate. Its purpose is to locate the OFDM symbol boundaries. The autocorrelation value is calculated by multiplying the I/Q samples values with delayed and conjugated I/Q samples and accumulation. A division by the energy normalizes the value. A peak is detected on the highest amplitude if more than 32 samples exceed a specified threshold and the distance to the last peak is more than 2,160 samples.

Another function of the radio frame synchronization block is the measurement and compensation of the CFO. The integer frequency offset (IFO) estimation is calculated based on the distance between the cross-correlation and the autocorrelation peak. The IFO is assumed to be zero only if it is below a certain threshold. The fractional frequency offset (FFO) is calculated based on the phase of the autocorrelation peak. The resulting CFO estimate is obtained by adding the IFO and FFO estimates. To prevent noisy estimates from sifting the estimate too much, the fractional part is multiplied with a *CFO factor* before it is used for updating the CFO estimate. When the synchronization is found, the new CFO estimate is only applied at the start of a radio frame. For debug purposes, a *static CFO* value can be configured which overrides the CFO estimation.

After multiple PSS signals are detected consecutively and the IFO estimation is complete, the Timing Adjustment block calculates the position of the start of the radio frame. The Radio Frame Alignment block uses this position to pass an entire time-aligned radio frame to the subsequent modules. The **FFT window timing advance** parameter can be used to set the number of samples that the receiver should cut into the cyclic prefix. This parameter together with the parameters mentioned before are part of the **sync configuration** cluster and can be set from the host.

If PSS or OFDM peaks are missing, the IFO Estimation block invalidates at least one radio frame of samples. In this case the samples are not passed to the subsequent modules.

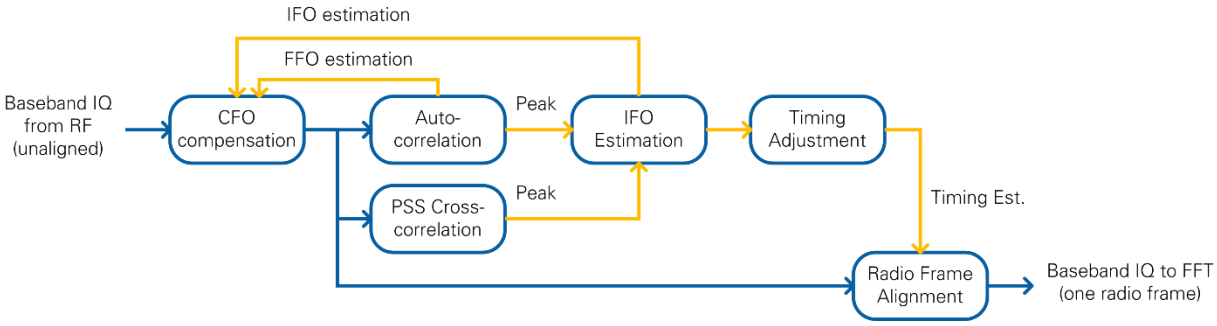

Figure 29: Radio Frame Synchronization Block Diagram

3.2.3.2 DL RX IQ Processing

This module reads the radio-frame aligned signal in time domain and outputs the channel-equalized subcarriers that are associated to the physical channels.

As shown in Figure 30, it includes the following functional blocks:

- CP removal
- FFT conversion
- Resource demapping
- CRS based channel estimation and equalization
- UERS based channel estimation and equalization

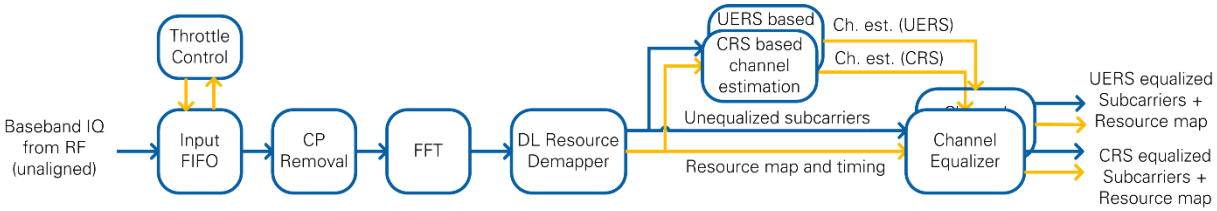

Figure 30: Block Diagram of DL RX IQ Processing

An internal FIFO is used to decouple the incoming samples from the rest of the processing chain. The throttle control module waits until enough samples for one complete OFDM symbol (FFT size + CP) are available before it passes them as a consecutive stream to the next modules.

The next module is the CP removal, which removes the valid flag from the samples belonging to the cyclic prefix. The 2,048 remaining samples are sent to a Xilinx FFT.

The output of the FFT are 2,048 the subcarriers in frequency domain. First, the resource mapper selects the 1,200 allocated subcarriers by removing the surrounding whitespace and the DC carrier in the center. Then it generates the timing information for each sample and the resource grid by marking each sample for its corresponding channel by using a Boolean cluster. The resource mapping is based on a fixed frame structure configuration described in the LTE specifications. All subsequent modules use this Boolean cluster with elements for each LTE channel to determine if this sample is relevant.

The FFT output data is fed into two separate channel estimation blocks running in parallel. The first channel estimation is based on the CRS. The channel estimate values are calculated by conjugate complex multiplications. A linear interpolation is applied in frequency domain between adjacent reference symbols, as shown in Figure 31. On the edges of the symbol, the nearest estimated value is replicated (zero order hold). OFDM symbols not containing CRS sequences rely on the last channel estimation (zero order hold in time, as shown in

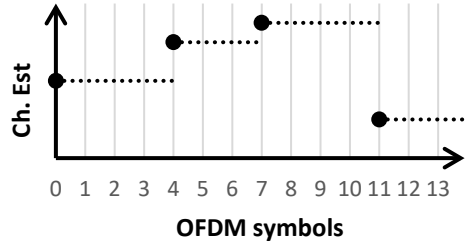

Figure 32).

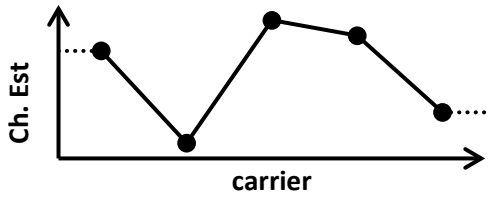

Figure 31: Channel Estimation over Frequency

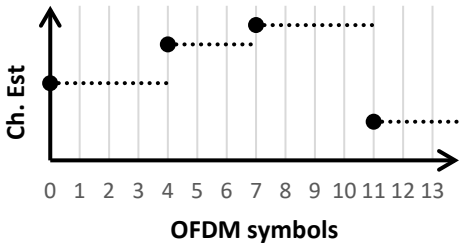

Figure 32: Channel Estimation over Time

The second channel estimation is based on the UERS. These pilots are transferred in each PRB assigned to the PDSCH addressed to the UE. At this point, there is no knowledge about the PRB allocation. Therefore, the channel estimation is done on each PRB. The same resource elements can use multiple UERS, so the averaging over one subframe is used in the time domain to cancel other possible sequences (multi-user interference cancellation). In the frequency domain, linear interpolation is used within the PRBs with zero-order hold applied at the edges.

The channel estimation is delivered sample by sample to the channel equalization modules parallel to the data. The channel equalization determines the result from the data sample *d* and the channel estimate *e* by using the following equation:

$$\frac{d \cdot e^*}{|e|^2} = \frac{d}{\sqrt{e \cdot e^*}} \cdot \frac{e^*}{\sqrt{e \cdot e^*}}$$

3.2.3.3 PDCCH Receiver

The PDCCH Receiver works on the output of the CRS-based channel equalization for the first OFDM symbol of each subframe (CFI fixed to 1). It decodes the DL control information (DCI) for the UE given in the PDCCH channel. The block diagram is shown in Figure 33.

The PDCCH receiver first extracts the PDCCH subcarriers from CRS equalized subcarriers by evaluating the channel map that is passed to the module in addition to the subcarrier I/Q data. An LLR demapper translates the symbols into soft bits, which are deinterleaved and descrambled by the given system and timing parameters.

After this step, the DCI demultiplexer extracts one DCI message from the PDCCH. The CCE offset parameter can be used to determine the DCI message location. The DCI decoder uses the Xilinx Viterbi Decoder Core to decode the DCI transport block bits from the given soft bits. The checksum of the CRC is calculated on the transport block. If this value matches the configured RNTI, the message is marked as valid.

The valid DCI message is interpreted according to the implemented DCI format (Figure 3). A validation module invalidates the DCI message in case the content is not supported, for example, when MCS > 28. It is also invalidated if a DL assignment is received outside a DL subframe.

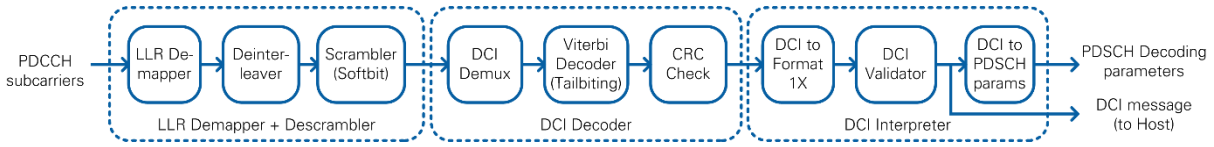

Figure 33: PDCCH Receiver Block Diagram

3.2.3.4 PDSCH Receiver

Depending on the system configuration, either CRS or UERS equalized QAM symbols are used. The selection is performed inside the PDSCH Sample Select module shown in Figure 34. This module buffers the incoming subcarrier data in FIFOs until a valid PDSCH Decoding Parameter configuration is received from the PDCCH Decoding. In case of CRS-equalized subcarriers, the first subframe is received before the PDSCH Decoder configuration. In case of UERS-equalized it is received later because of the higher latency of the UERS channel estimation. The PDSCH subcarriers provided by the PDSCH FIFO Control module comprise the

PDSCH subcarriers of all 100 resource blocks. Another resource demapper marks the PDSCH QAM symbols as valid or invalid based on the RB allocation from the decoded DCI message that is provided as part of the PDSCH Decoding parameters.

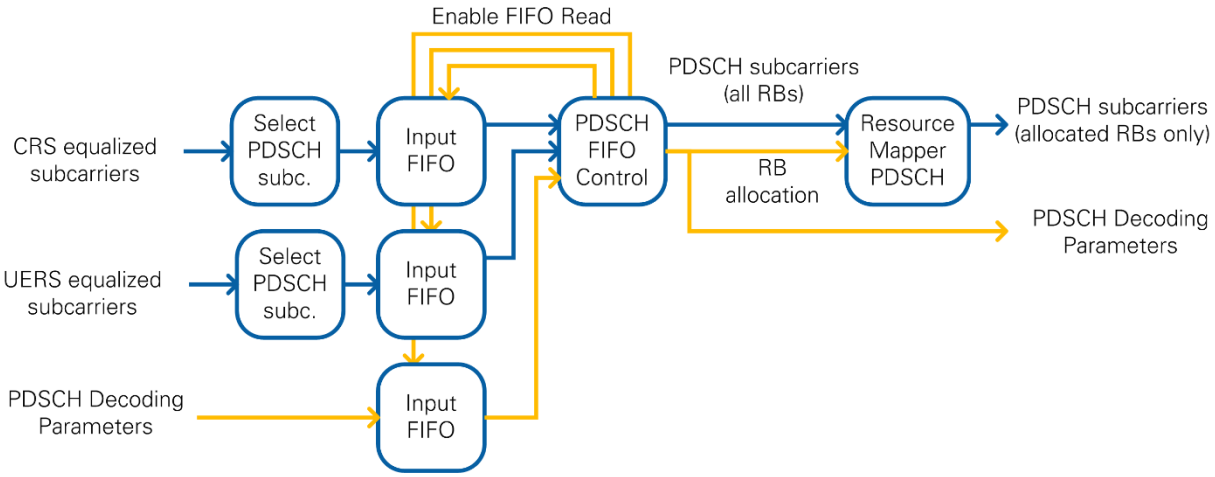

Figure 34: PDSCH Sample Select Block Diagram

The selected PDSCH subcarriers and the corresponding PDSCH decoding parameters are passed to the PDSCH Bit Processing module shown in Figure 35.

The valid symbols are interpreted as softbits in the LLR demapper. These softbits are descrambled using the cell parameters and the RNTI. A Scrambler initialization module calculates the initialization value of the shift register which corresponds to 1,600 iterations in just 32 clock cycles.

The PDSCH transport blocks are decoded inside the PDSCH Channel Decoder. It includes the Xilinx LTE UL Channel Decoder core which is configured to perform turbo decoding. The necessary parameters (TB size, code block size, number of code blocks) are calculated on the FPGA and based on the received DCI message and the configured system settings. The hard bits from the decoder are transferred to the host using a dedicated FIFO. Another FIFO writes the decoding status information to the host.

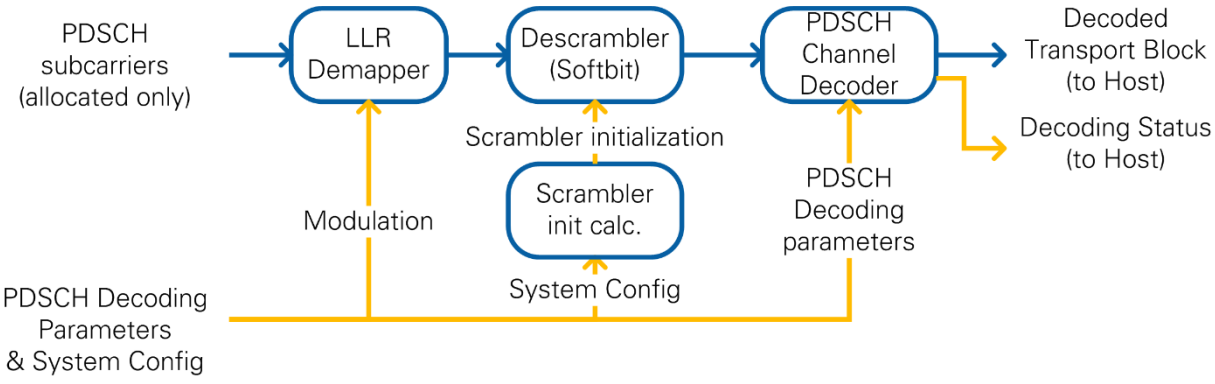

Figure 35: PDSCH RX Bit Processing Block Diagram

3.2.4 UL Transmitter

The UL Transmitter is implemented by the FPGA top-level UE. In the block diagrams of Figure 19, it corresponds to the UL TX PHY block.

As shown in the simplified block diagram in Figure 36, it performs the following tasks:

- PUSCH encoding
- Mapping to resource elements
- IFFT conversion + CP conversion (OFDM modulation)

The following reference signals and physical channels are mapped:

- Demodulation reference signal (DMRS)
- Sounding reference signal (SRS), if enabled
- Physical UL shared channel (PUSCH)

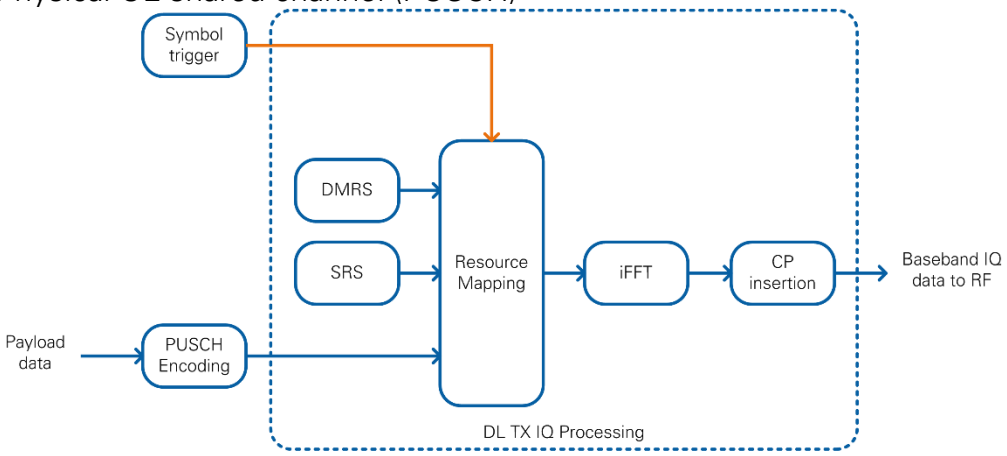

Figure 36: Simplified Block Diagram of UL Transmitter

Channel Encoding is performed for the PUSCH as shown in Figure 37. The encoding is performed for each UL subframe by the FPGA implementation which allows real-time operation. The other signals are read out from LUTs.

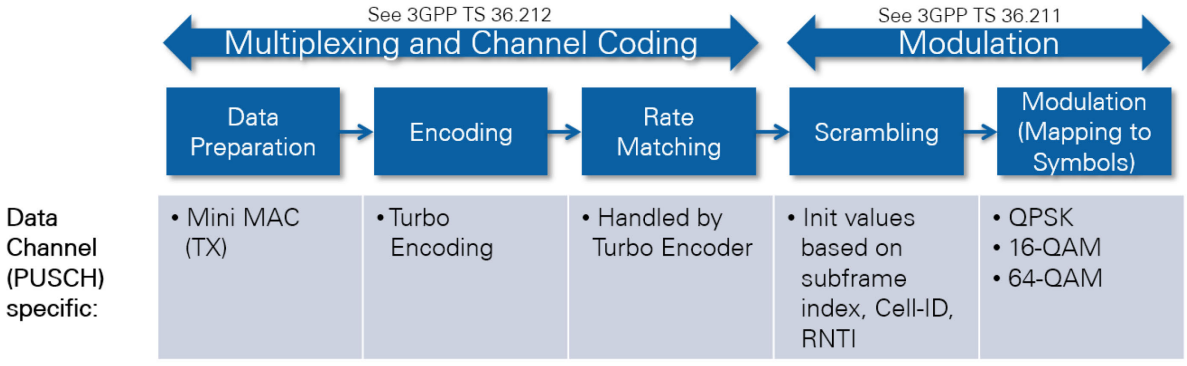

Figure 37: Channel Encoding Performed in the UL Transmitter

3.2.5 UL Receiver

The UL RX is implemented by the FPGA top-level variant eNodeB. As shown in the simplified block diagram in Figure 38, it performs the following tasks:

- CP removal + FFT conversion (OFDM demodulation)
- Demapping of the resource elements to the different physical channels
- DMRS-based channel estimation and equalization
- PUSCH decoding

The following reference signals and physical channels are demapped:

- DMRS
- SRS, if enabled
- PUSCH

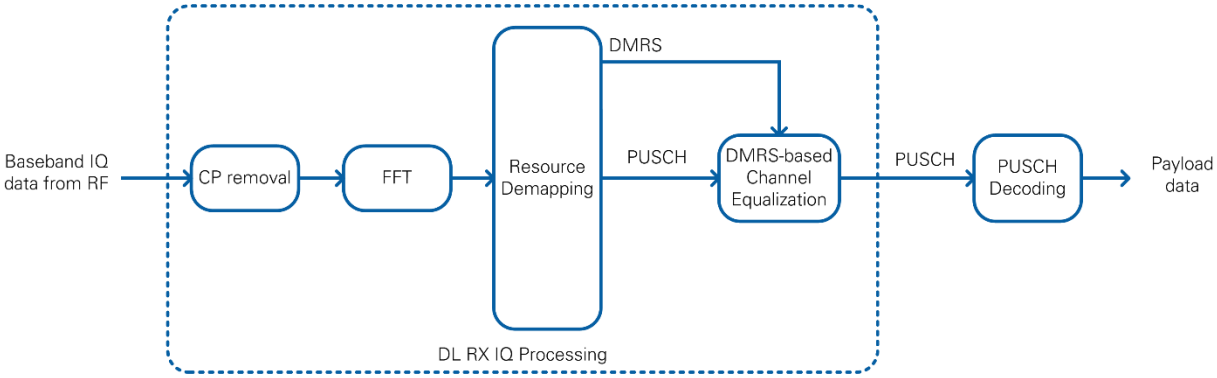

Figure 38: Simplified Block Diagram of the DL RX

The UL RX receives the I/Q samples in time domain from the RF loop derived from the sample streaming project. It performs the I/Q impairments correction, digital down conversion, frequency shift, and the down conversion from the ADC sample rate to the LTE sampling frequency of 30.72 MHz.

In contrast to the DL RX, synchronization is not performed because the UE is required to send the UL subframe with the correct timing. Therefore, the incoming I/Q samples are already time-aligned.

The DMRSs are used for channel estimation and equalization.

The PUSCH decoder decodes the PUSCH. The PUSCH configuration parameters are calculated based on the MCS and RB allocation parameters set from the host. The decoded PUSCH transport blocks are written to the host using a target-to-host DMA FIFO.

The submodules used for the UL RX are very similar to the submodules used for the DL RX. Refer to section 3.2.3 for more details about the implementation of these submodules.

3.2.6 Clocking Considerations

The following three main clock domains are used inside the FPGA: 40 MHz onboard clock, 120 MHz, 130 MHz, or 200 MHz Sample clock, and 192 MHz baseband clock.

The configuration loops are connected to the 40 MHz clock domain. The configuration information is set prior to execution and used as constants elsewhere in the design.

All LTE baseband processing loops run at a clock rate of 192 MHz. The ADC and DAC interfaces run at Sample clock rate, in addition to the sample rate converters that create the 30.72 MS/s I/Q data. The 192 MHz processing clock and Sample clock are not synchronized, and this difference is accounted for in the design. The processing done in the 192 MHz domain has enough margin to account for frequency tolerances between the 192 MHz clock and the Sample clock.

The DL TX chain uses a synchronization mechanism to keep the baseband processing aligned and to avoid any under- or overflows of the FIFO that transfers data between the two clock domains. The sample clock domain is used as the absolute time reference. A trigger is generated in the sample clock domain every 10 milliseconds (for each radio frame). This trigger is sent to the 192 MHz processing domain to initiate the creation of a new radio frame. The FIFO between the two clock domains guarantees a constant data rate for the digital up conversion module.

For testing purposes, there is an internal loopback FIFO on the DL FPGA that you can use to bypass the RF by directly transferring samples from the DL TX to the DL RX baseband processing. The internal loopback is disabled by default and can be enabled from the host.

3.3 Host Implementation

The application framework provides three sample host implementations that cover all important features of the application framework. As described in section 2.1, three different host implementations are provided:

- DL: Establishes a DL link in either a single-device setup or a double-device setup
- eNodeB: Provides the eNodeB side in a double-device setup and implements the DL TX and the UL RX of an eNodeB
- UE: Provides the UE side in a double-device setup and implements the DL RX and the UL TX of a UE

Each host implementation interfaces with the bitfile that was built from the corresponding FPGA implementation. It demonstrates the main functionalities for each implementation. This functionality includes configuration of the FPGA target, exchanging payload data, and monitoring the system status.

As shown in the schematic overview of Figure 39, each host implementation is split into an Initialization part, several processing loops, and a Cleanup part. Figure 40 shows a screenshot of the LabVIEW G code.

All parts and processing loops are further described in the next sections.

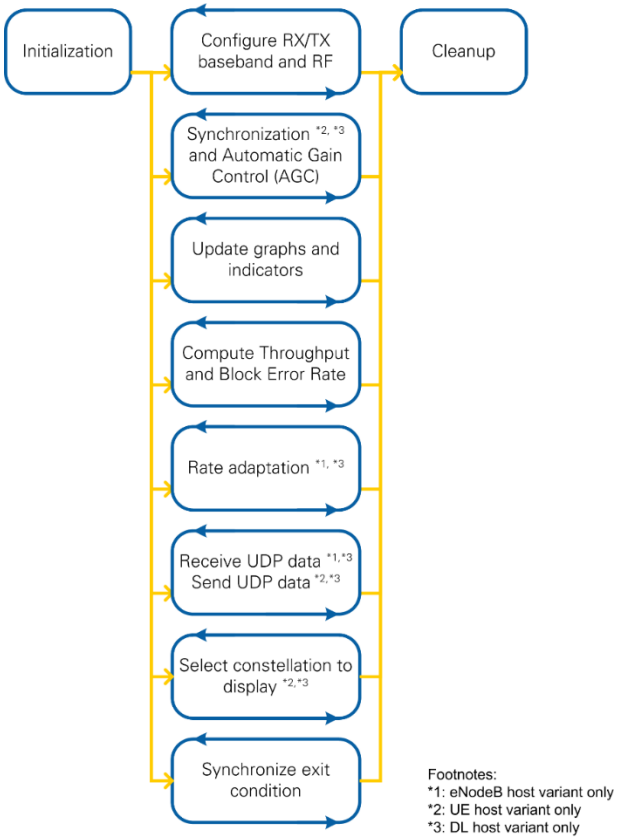

Figure 39: Host Block Diagram—Schematic Overview

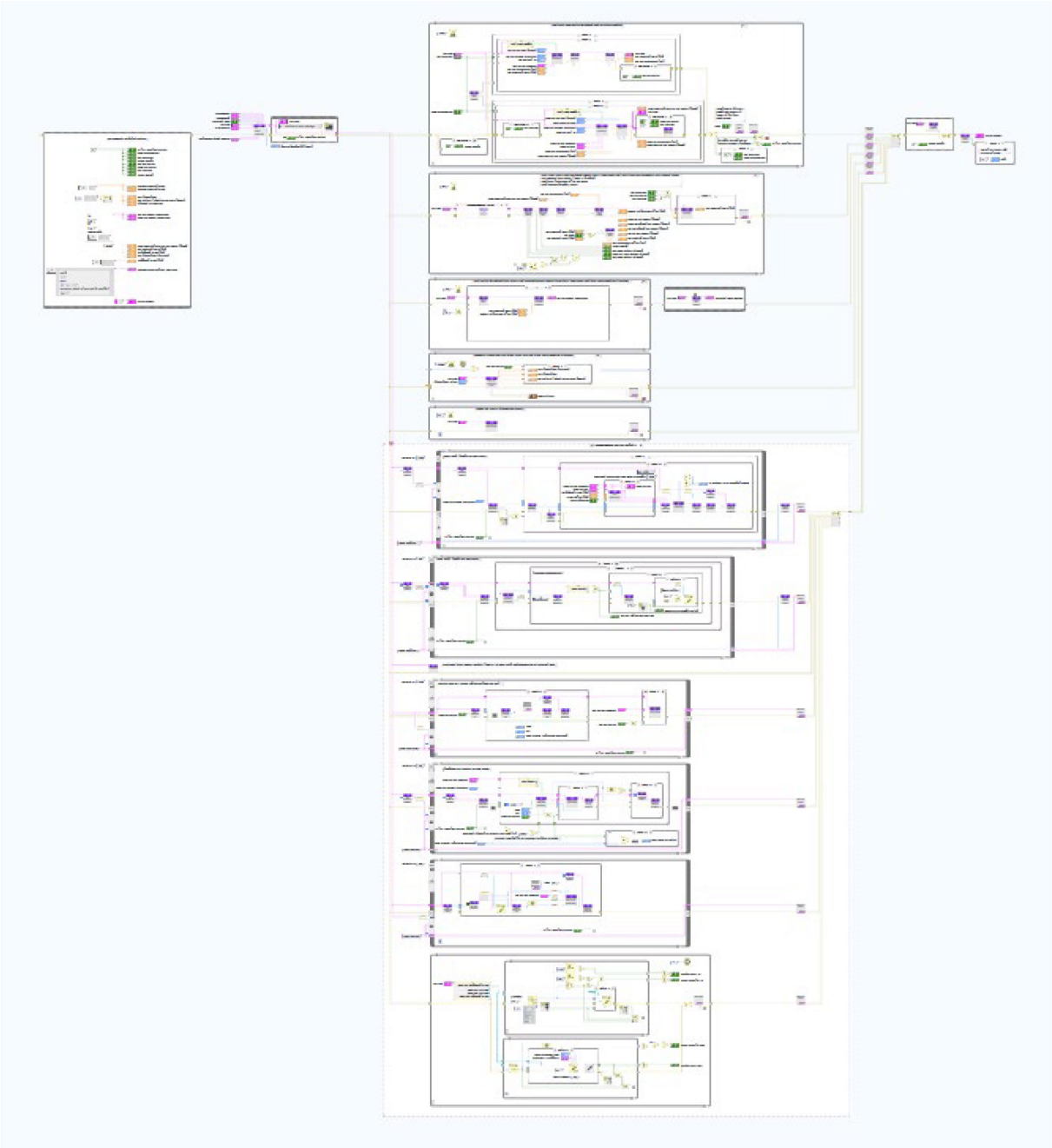

Figure 40 Host Block Diagram—Screenshot of LabVIEW G code

3.3.1 Initialization, Synchronize Exit Condition and Cleanup

The entry point of the code is the initialization block. It sets several controls and indicators to default values. Also, it prepares the session cluster by starting the necessary queues and loading the FPGA bitfile to the configured device. All processing loops use this session cluster during execution to exchange data or to access the FPGA resources.

All processing loops are implemented as while loops that run in parallel during the execution of the host VI. A dedicated *stop* queue is used to synchronize the stop condition across all loops. The *synchronize exit condition* loop checks it when the

Stop button is pressed and accordingly sets the stop condition. The stop condition is also set in case an error occurred in any of the processing loops.

After all processing loops are stopped, the handles from the session are closed; that is, the queues are stopped and the FPGA reference is closed.

3.3.2 Configure RX/TX Baseband and RF

This loop handles the configuration of the target specific RF and the LTE processing chains. After changes to the RX or TX enable switches, the required parameters are passed to the LTE FPGA processing chain using the settings presented on the front panel, that is, the MCS and the resource block allocation are configured when the DL TX is enabled. After the LTE processing parameters were written to the FPGA, the RF chain is configured and started. Some error cases are caught and presented to the user in dialog boxes.

3.3.3 Synchronization and Automatic Gain Control

The loop continuously monitors the received signal power and adjusts the gain for the RX path accordingly. It reports the status of the radio frame synchronization on the FPGA and updates the overflow indicators on the front panel.

This loop also reads the PDSCH decoding status (DL, UE host variants) or the PUSCH decoding status (eNodeB host variant). This information is needed for the outgoing UDP Stream and the throughput calculation. Therefore, the elements that are read from the PDSCH/PUSCH decoding status FIFO are duplicated and written to multiple queues.

3.3.4 Update Graphs and Indicators

This loop reads and processes status information from the FPGA and updates the associated graphs and indicators on the host front panel. For example, it reads the baseband signal of the associated RX or TX processing chain, calculates the power spectra, and updates the corresponding graphs. This loop also updates the constellation diagram shown on the currently selected tab. For the DL and UE host variant, it also reads the current channel estimation, calculates both the subband and wideband SINR from it, and updates the associated graphs and indicators.

3.3.5 Compute Throughput and Block Error Rate

This loop calculates the throughput and the BLER based on the state of several queues filled by the other processing loops.

The queues are emptied on a fixed time basis (1,000 ms), and the elements are accumulated as follows:

$$Throughput_{User\ data} = \sum_{1000\ ms} n_{payload\ bits}$$

$$Throughput_{PDSCH\ (CRC\ ok)} = \sum_{1000\ ms\ |CRC\ ok} TBSize$$

$$Throughput_{PDSCH\ (overall)} = \sum_{1000\ ms} TBSize$$

Where $n_{payload\ bits}$ is the number of the payload bits per received transport block.

This value is the result of the MiniMAC header interpretation. $TBSize$ is the size of the received transport block. The **PDSCH (CRC ok)** throughput value considers only the transport blocks that were received successfully (that is, without CRC errors). The **PDSCH (overall)** throughput accumulates the sizes of all transport blocks independently on the CRC status.

The sync failure rate, $P_{Sync\ Failure}$, and the PDCCH and PDSCH BLER $BLER_{PDCCH}$ and $BLER_{PDCCH}$ are calculated as follows:

$$P_{Sync\ Failure} = 1 - P_{Sync\ Successful} \quad , \qquad P_{Sync\ Successful} = \frac{n_{Sync\ Successful}}{n_{Sync\ Checked}}$$

$$BLER_{PDCCH} = 1 - P_{Sync\ Successful} \cdot P_{PDCCH\ Successful}$$

$$P_{PDSCH\ Successful} = \frac{n_{PDSCH,CRC\ ok}}{n_{PDSCH}}$$

$$BLER_{PDSCH} = 1 - P_{Sync\ Successful} \cdot P_{PDCCH\ Successful} \cdot P_{PDSCH\ Successful}$$

$$P_{PDCCH\ Successful} = \frac{n_{PDCCH,CRC\ ok}}{n_{PDCCH}}$$

Where $n_{Sync\ Successful}$ is the number of times the sync was successful and $n_{Sync\ Checked}$ is the number of times the sync status was checked (inside the synchronization and automatic gain control loop).

n_{PDCCH} is the number of decoded DCI messages and $n_{PDCCH,CRC\ ok}$ is the number of messages that were decoded successfully (that is, without CRC errors).

n_{PDSCH} is the number of decoded PDSCH transport blocks and $n_{PDSCH,CRC\ ok}$ is the number of PDSCH transport blocks that were decoded successfully (that is, without CRC errors).

3.3.6 Update Dynamic DL Configuration (Includes Rate Adaptation)

The DL and the eNodeB host variants contain a mechanism that updates the dynamic DL configuration upon receiving a timing indication from the TTI handling module. The message sequence chart is shown in Figure 41.

1. Upon receiving a subframe trigger, the TTI handling module writes a *Timing indication* for the next subframe to the host using the *T2H Timing Indication* target-to-host DMA FIFO.
2. The host reads the **timing indication**.
3. The host determines the new **DL TX configuration**.
4. The host writes the updated **DL TX configuration** to the TTI handling module using the *H2T DL TX Dynamic Configuration* host-to-target DMA FIFO.
5. Upon the next subframe trigger, the TTI handling module reads the updated DL TX configuration and passes it to the PDCCH/PDSCH Encoding modules.
6. Repeat steps 1 through 5.

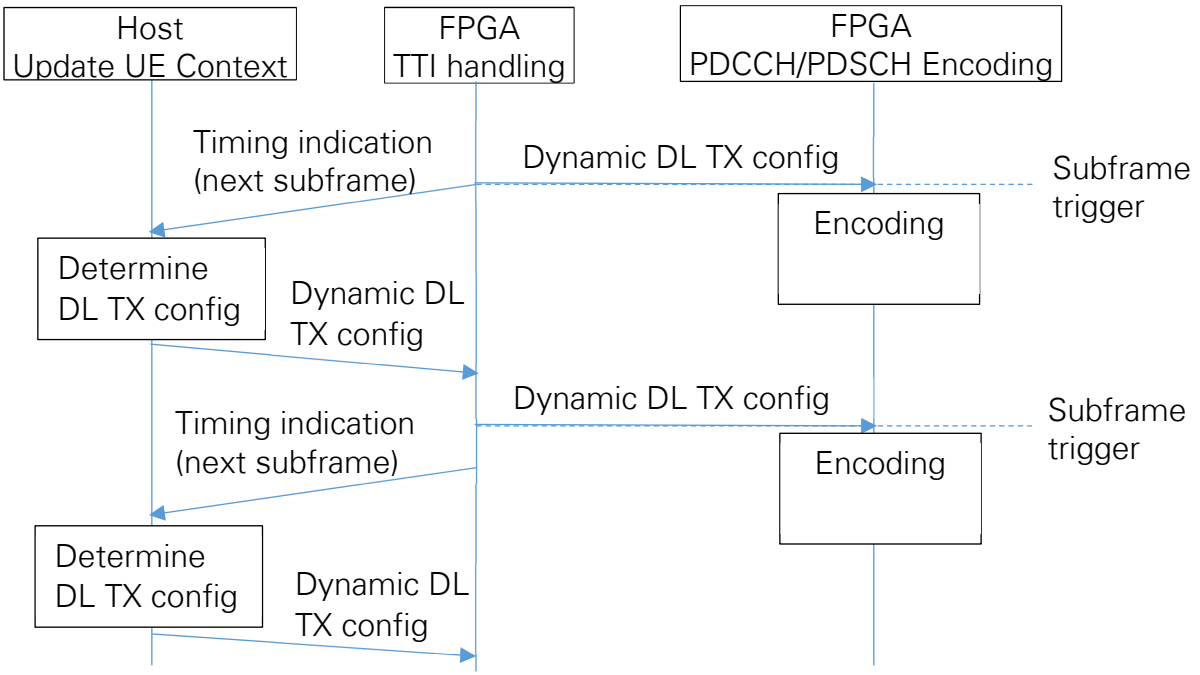

Figure 41 - Sequence Chart for Updating Dynamic DL Configuration

The message sequence chart assumes ideal timing where the total latency from writing the timing indication until reading the dynamic TX configuration is smaller than one millisecond. This strict regular timing can only be achieved when deploying the host code on a real-time target. When deploying the host code on a Windows operating system, there will be additional jitter which will cause timing indications to be received too late and as a result the dynamic DL TX configuration will also be received too late on the FPGA. To compensate for these timing effects, the TTI Handling module has two modes:

- RT Target=true: The TTI handling module assumes that the dynamic DL TX configurations are received in time (within one millisecond). If one is missing, an empty configuration will be provided to the PDCCH/PDSCH encoding modules.
- RT Target=false: The TTI handling module is tolerant for dynamic DL TX configurations to be missing or received too late. In these cases, the

previously received dynamic DL TX configuration will be provided to the PDCCH/PDSCH encoding modules.

By default, RT Target is set to false which makes the TTI handling module tolerant for dynamic DL TX messages to be missing or received too late.

The update dynamic UE contexts loop on the host also implements a rate adaptation functionality which sets MCS depending on the reported wideband SINR value. This value is either read directly from the FPGA (DL host variant) or received from the UE as part of the UL feedback (eNodeB host variant). The implemented SINR-MCS mapping table is calibrated so that the resulting PDSCH BLER is around 10% if no offset is applied. Use the **SINR Offset [dB]** control to apply an SINR offset which can achieve a lower PDSCH BLER value.

3.3.7 Receive UDP Data and Send UDP Data

In the eNodeB and DL host variant, this loop handles incoming UDP data, which is used as the payload data for DL TX. The data is provided by an external application and read from the port number specified on the front panel. The data is pushed to the FPGA using a host to target FIFO.

In the UE and DL host variant, this loop also handles outgoing UDP data. This data stream represents the payload data which was received and successfully decoded from the DL RX. After the data was transferred to the host using a target to host FIFO, the data is packed into a UDP stream and sent to an external application. The IP address and port number can be specified on the front panel.

3.3.8 Select Constellation to Display

For the UE and DL host variant, this loop checks which I/Q constellation is shown depending on the currently displayed tab. The associated physical channel is determined and requested from the FPGA. This step is necessary because a shared target-to-host DMA FIFO is used to transfer the I/Q samples to the host.

3.4 Real-Time (RT) Implementation Overview

3.4.1 Transport Layer

The application framework uses Linux pipes as the transport layer. Linux pipes allow fast and reliable transport. They also allow L2 implementations in LabVIEW as well as in other languages that support access to Linux pipes such as C, C++ or Python.

The LabVIEW implementation encapsulates the Linux pipes with RT FIFOs. This allows an easy replacement of the Linux pipes with other transport mechanism as well as bypassing the transport for inner LabVIEW L2 implementations.

Figure 42, Figure 43 and Figure 44 show the data flows through the transport layer. The data flow does not depend on whether data is transmitted from L1 to L2 or the other way. Also, for the LabVIEW application there is no difference whether the data is sent to or comes from a LabVIEW instance or an external application.

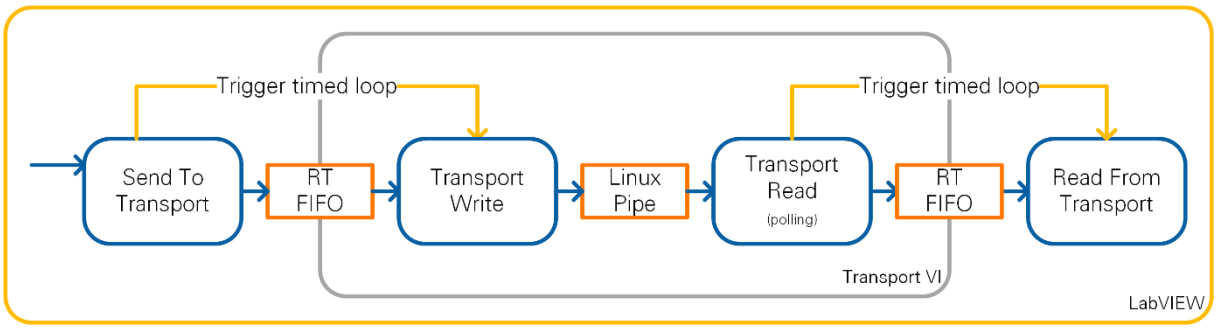

Figure 42 Data Transport (LabVIEW to LabVIEW)

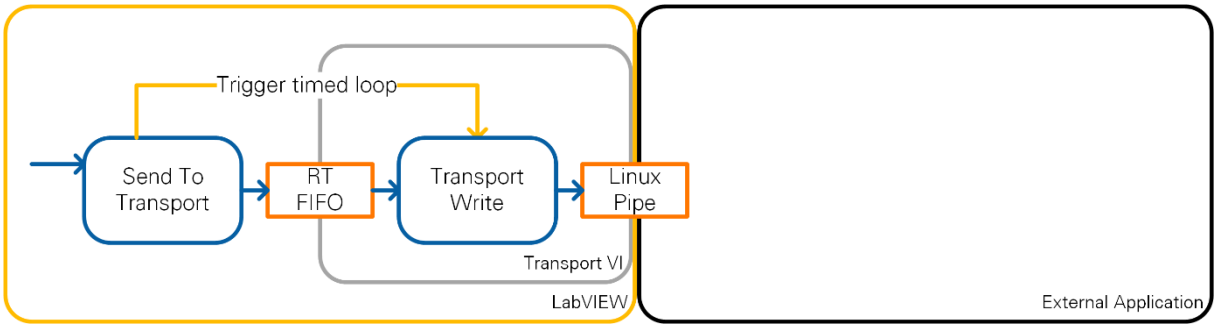

Figure 43 Data Transport (LabVIEW to External Application)

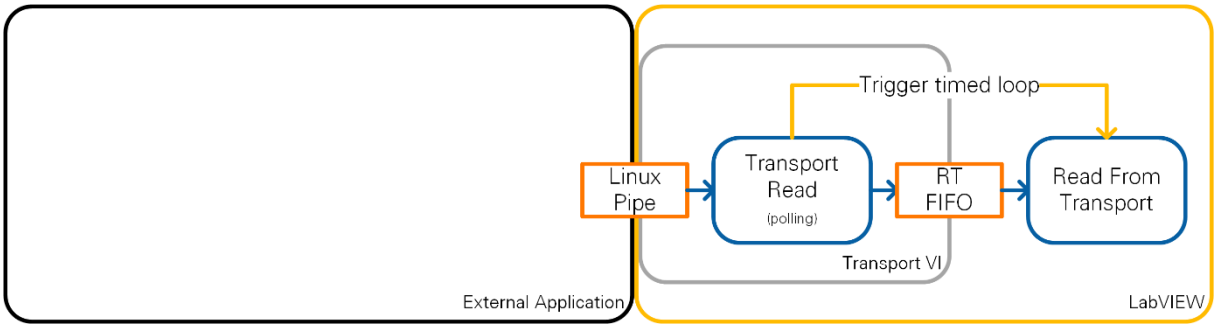

Figure 44 Data Transport (External Application to LabVIEW)

3.4.2 Data Flow

Because Linux pipes are unidirectional at least two pipes are needed to do bidirectional communication between MAC and PHY. Due to the nature of pipes (first in first out), different pipes needs to be defined to prioritize one message over the other. The application framework defines three pipes for data exchange to be able to prioritize messages from PHY to MAC.

- Pipe one is reserved for highly prioritized messages from PHY to MAC. These are TTI trigger messages.
- Pipe two is reserved for messages sent from MAC to PHY. This includes the PHY TX request messages.
- Pipe three is reserved for normal prioritized messages sent from PHY to MAC. This includes the PHY_CNF messages created as a response to PHY TX requests as well as the PHY RX indication messages.

Figure 45 shows how messages are assigned to the data pipes for the three different scenarios defined in 3.3.

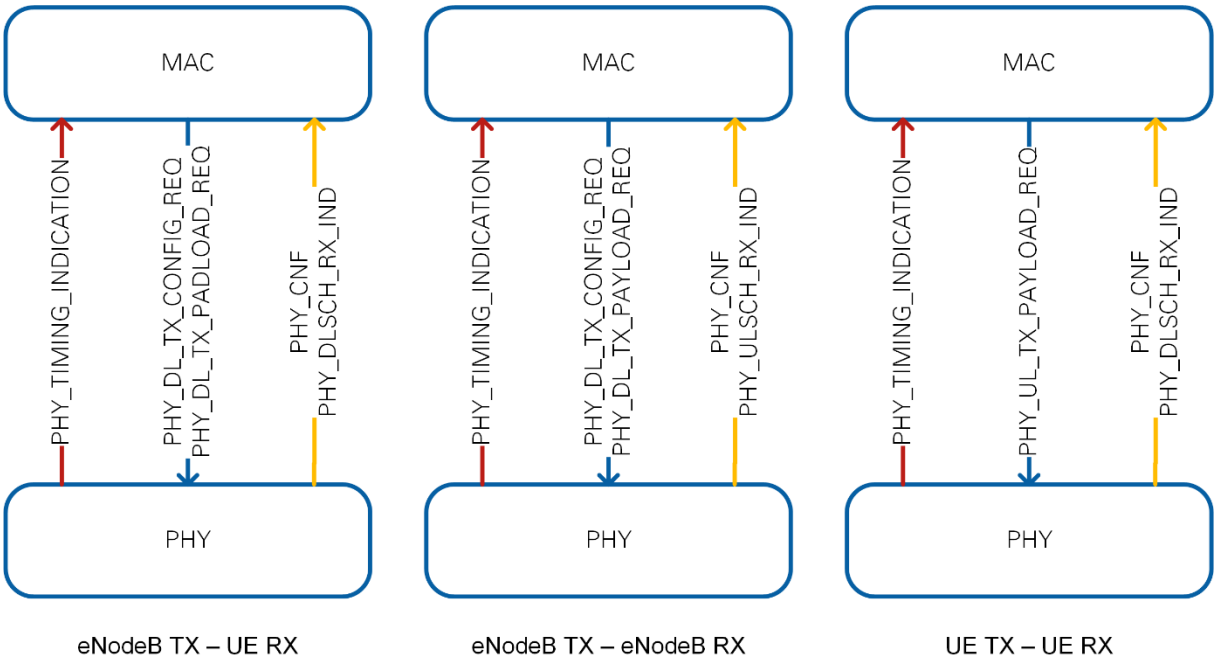

Figure 45 Assignment of Messages to Pipes

3.4.3 Data Handling Loops

In each scenario defined in section 3.3, the application framework defines five timed loops. The loop names used here corresponds to the *structure in* of the timed loops on the LabVIEW block diagram.

On the PHY layer there are three loops (in decreasing priority):

- PHY SAP Tim
  - Reads TTI indications from DMA FIFO
  - Passes TTI indications as messages into first pipe
- PHY SAP Tx
  - Reads PHY TX requests from second pipe
  - Parses and evaluates PHY TX requests
  - Generates PHY CNF messages
  - Passes PHY CNF messages to third pipe
  - Configures TX according to PHY TX configuration requests
  - Passes payload to DMA FIFO according to PHY TX payload requests
- PHY SAP Rx
  - Reads PHY RX indications
  - Passes PHY RX indications into third pipe

On the MAC layer there are two loops (in decreasing priority):

- MAC Stub Tx
  - Reads TTI indication from first pipe
  - Generates PHY TX requests for configuration (DL only) and payload
  - Passes PHY TX requests to second pipe
  - Saves sent PHY TX requests in RT FIFO for later comparison with received PHY CNF messages

- MAC Stub Rx
    - Reads messages from third pipe
    - If message is a PHY CNF message:
        - Reads corresponding PHY TX request information from RT FIFO written by the first MAC loop
        - Checks PHY TX request message versus PHY CNF message
    - If message is a PHY RX indication:
        - Decodes the RX indication
        - Passes decoded payload for further MAC processing

Loops that read data from Linux pipes through RT FIFOs (second PHY loop, MAC loops) only run if they are triggered by the transport layer. The other loops are triggered when data is available in DMA FIFOs.

The application framework uses different handlers to process messages. Each handler receives a context (eNodeB or UE). The context decides to process or create messages, whereas the overall logic remains the same regardless of the context.

- PHY TX Request Confirm Abstraction: Create expected PHY CNF messages for a TX request. Contains two PHY CNF messages for eNodeB context (configuration and payload) or one PHY CNF message for UE context (payload only).
- PHY Receive RX Indication: Decodes either PHY UL RX indication (eNodeB context) or PHY DL RX indication (UE context).
- PHY TX Request Confirm Handler: Decodes PHY TX request messages and prepares the message content for further processing by PHY FPGA. It furthermore generates PHY CNF messages.

4 Conclusion

LabVIEW Communications LTE Application Framework provides a real-time LTE DL and UL including basic feedback mechanisms running on NI SDR hardware. This framework enables you to focus on a specific area of research by utilizing the existing link and only making changes or additions where desired.

Because of the flexibility of LabVIEW and the modularity of the framework, you can easily exchange portions of the design for prototyping new algorithms for future wireless systems. In addition, LabVIEW's native interface between the host and the FPGA means that the design can be partitioned to profit from the parallel execution on the FPGA as well as calculations on the host.

The FPGA bitfiles shipped with the design are fully functional and support test modes with and without the RF. In addition, you can use external RF equipment, such as the PXIe-5644 Vector Signal Transceiver, to simulate interference and various channel conditions.

This application framework offers a variety of starting points for wireless research and prototyping. Start now by downloading an evaluation copy of LabVIEW Communications at www.ni.com/labview-communications.

Questions? Email us at labview.communications@ni.com.

5 Appendix

5.1 LTE Application Framework PHY SAP Message Definitions

The PHY SAP defines a set of messages necessary to exchange payload data between MAC and PHY.

Figure 46 shows messages and corresponding sub-messages clustered for eNodeB and UE specific messages.

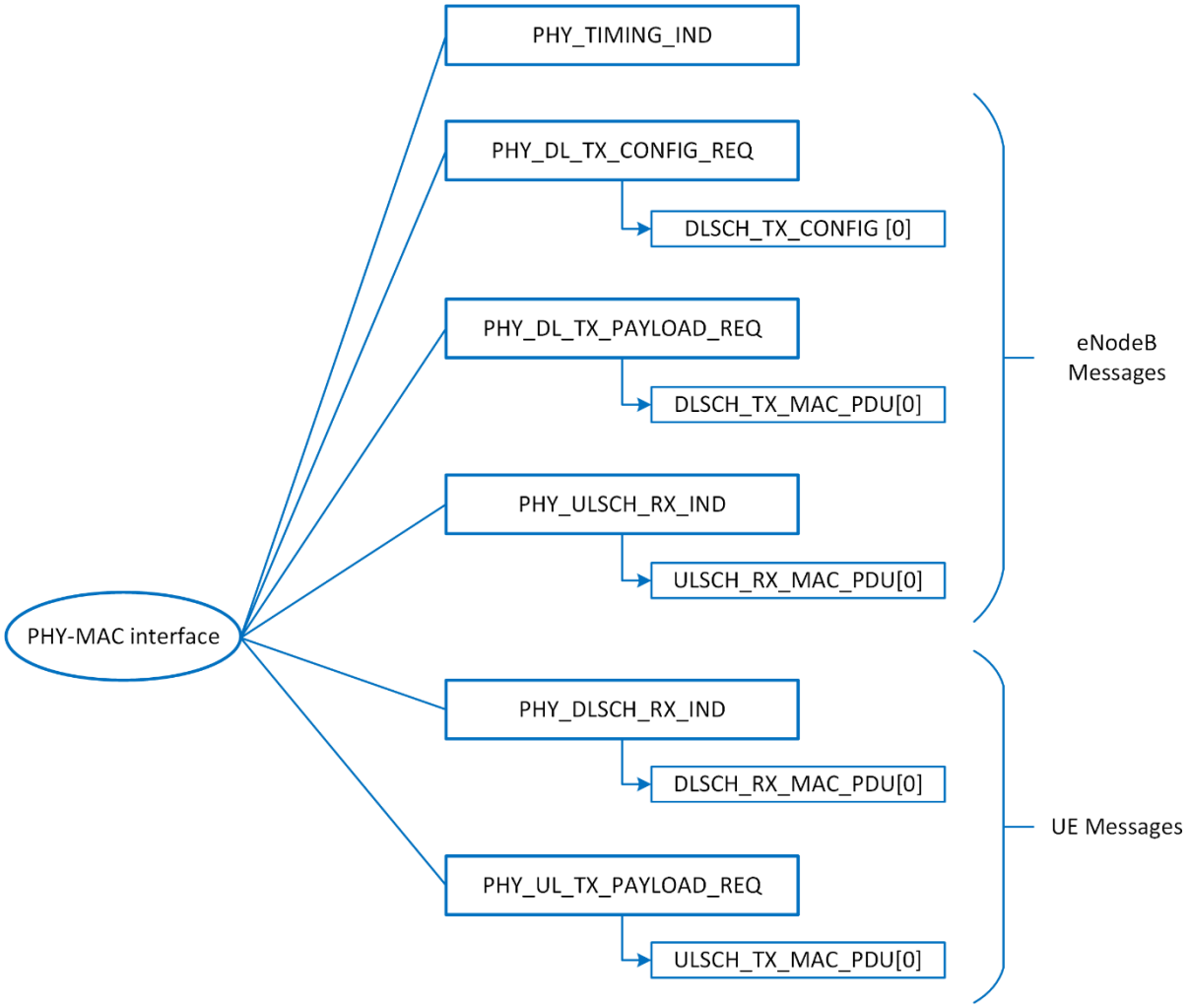

Figure 46 Messages and Hierarchy at PHY-MAC Interface

5.1.1 Message Numbering Schema

The following table shows usage of message bits in the message number. Values in bold are used by the actual implementation.

Message ID bits	Usage / Meaning
b15 b14	SAP (service access point) interface type 00: layer control interface (from/to RRC) **01: data path interface** 10: reserved 11: reserved (for example, for debug interface)
b13 b12	Service layer ID 00: PHY **01: MAC** 10: RLC 11: PDCP
b11 b10	Location / Scope 00: common **01: eNodeB** **10: UE** 11: reserved
b9 b8	Message type **00: Indication (IND)** **01: Request (REQ)** **10: Confirm (CNF)** 11: reserved
b7…b0	Message ID within category

5.1.2 Used Message IDs

The application framework uses the following message IDs

Message name	Message ID (binary) b15 … b0	Message ID (hex)
PHY_TIMING_IND	0100 0000 0000 0001	0x4001
PHY_UL_TX_PAYLOAD_REQ	0100 1001 0000 0010	0x4902
PHY_UL_TX_PAYLOAD_CNF	0100 1010 0000 0010	0x4A02
PHY_DLSCH_RX_IND	0100 1000 0000 0001	0x4801
PHY_DL_TX_CONFIG_REQ	0100 0101 0000 0001	0x4501

Message name	Message ID (binary) b15 … b0	Message ID (hex)
PHY_DL_TX_CONFIG_CNF	0100 0110 0000 0001	0x4601
PHY_DL_TX_PAYLOAD_REQ	0100 0101 0000 0010	0x4502
PHY_DL_TX_PAYLOAD_CNF	0100 0110 0000 0010	0x4602
PHY_ULSCH_RX_IND	0100 0100 0000 0001	0x4401

5.1.3 General Message Header (Applies to All Messages)

The general message header is prepended to all messages and contains the following fields.

Field	Type/ Length	Value range	Description	32-bit words
message type ID	U16		Message type identifier	2xU32
reference ID	U16	**Linear Counter**	Reference identifier used for mapping requests to the corresponding confirm messages and/or detection of lost indications or requests. For requests and indications, the content must be generated by a message type specific counter which starts at zero and is incremented with every newly generated request or indication of the specific type. For confirm messages, this field is field filled with the same reference ID as used in the handled request.	
instance ID	U8	0	Instance identifier (for example, for supporting parallel API connections of same type)	
message body length	U24		Message body length in bytes without any padding bytes which might be potentially added by the	

Field	Type/ Length	Value range	Description	32-bit words
			physical transport layer of a specific API implementation	

5.1.4 SAP Sub-header

This header is common to all messages. The confirm mode only applies for request messages. Other messages treat this as reserved field.

Field	Type/ Length	(Sup-ported) Value range	Description	32-bit words
SFN	U16	**0…1023**	System frame number. Depending on the specific message type it will be the SFN related to the reception of an event or the target transmission SFN.	**2xU32**
TTI index	U8	**0…9**	TTI index. Depending on the specific message type, it will be the TTI index related to the reception of an event or the target transmission TTI index.	
number of sub-messages	U8		Number of sub-messages contained in the hierarchical message	
confirm mode	U8	{0; 1}	0: no confirmation required **1: standard confirmation required**	
Reserved	U24	0	Reserved field (needed to ensure U32 alignment)	

5.1.5 Messages

5.1.5.1 PHY_TIMING_IND

This indication is used by the L1/PHY to indicate the start of a new TTI. Timing reference is the L1 DL**.** Therefore, the PHY_TIMING_IND at the eNodeB indicates the start of the DL signal transmission, and the PHY_TIMING_IND at the UE indicates the start of the DL signal reception.

SFN content is derived from L1 internal counter and used as basis for L2 on the eNodeB. On UE the SFN content is derived after initial DL sync. The L1 sends PHY_TIMING_INDs with SFN derived from an L1 internal counter (not aligned to the network SFN) but reset after every new sync success. Setting the L1 SFN counter from L2 is not possible. The L2 shall synchronize its SFN on UE and eNodeB side based on the PHY Timing IND messages. At the UE after L1 is synced to DL, L2

expects the PHY_TIMING_IND periodically, that is, for every TTI. This is the trigger for UE L2 that connection is established. UE L2 does not need to be informed about fine timing updates on the UE L1/PHY.

Field	Type/ Length	Value range	Description	32-bit words
message type ID	U16	**0x4001**	message type identifier	2xU32
reference ID	U16		Refer to section 3.4	
instance ID	U8		Refer to section 3.4	
message body length	U24	**8**	message body length in bytes	
SFN	U16		Refer to section 5.1.4	2xU32
TTI index	U8		Refer to section 5.1.4	
number of sub-messages	U8	**0**	Number of sub-messages contained in the hierarchical message	
Reserved	U32		Refer to section 5.1.4	

5.1.5.2 PHY_DL_TX_CONFIG_REQ

This message is used by the eNodeB L2/MAC to configure/request DL transmissions on the L1/PHY or the request changes of parameters which are related to DL transmissions.

Field	Type/ Length	Value range	Description	32-bit words
message type ID	U16	**0x4501**	Message type identifier	2xU32
reference ID	U16		Refer to section 3.4	
instance ID	U8		Refer to section 3.4	
message body length	U24		Message body length in bytes	
SFN	U16		Refer to section 5.1.4	2xU32
TTI index	U8		Refer to section 5.1.4	
number of sub-messages	U8		Number of sub-messages contained in the hierarchical message	
confirm mode	U8		Refer to section 5.1.4	

Field		Type/ Length	Value range	Description	32-bit words
Reserved		U24		Refer to section 5.1.4	
For number of sub-messages (in PHY_DL_TX_CONFIG_REQ)					
	sub-message type (parameters set)	U8	**{**0**;1;2;**3**}**	Sub-message type defined for PHY_DL_TX_CONFIG_REQ 0: DL_TX_COMMON_CONFIG **1: DLSCH_TX_CONFIG** **2:DCI_TX_CONFIG_DL_GRANT** 3:DCI_TX_CONFIG_UL_GRANT	
	parameter set ID (parameter set version)	U8		Refer to section 3.6	
	parameter set body length	U24		Parameter set body length in bytes (without the fields **sub-message type**, **parameter set ID**, and **parameter set body length**)	
	Parameter set body	<Parameter set length>		Parameter set body. Content of the specific sub-message according to the selected parameter set.	

5.1.5.2.1 DLSCH_TX_CONFIG

Field		Type/ Length	Value range	Description	32-bit words
sub-messages of PHY_DL_TX_CONFIG_REQ					
	sub-message type (parameters set)	U8	**2**	Sub-message type defined for **2: DLSCH_TX_CONFIG**	
	parameter set ID (parameter set version)	U8		Refer to section 3.6	
	parameter set body length	U24	**8**	Parameter set body length in bytes. Without the fields **sub-message type**, **parameter set ID**, and **parameter set body length**).	

Field		Type/ Length	Value range	Description	32-bit words
	PDU index	U8		Index to associate a transport channel configuration sub-message with the corresponding MAC PDU in the DL_TX_PAYLOAD_REQ o incremented with each transport channel configuration sub-message included in the DL_TX_CONFIG_REQ o reset to 0 with every new subframe or TTI	
	RNTI	U16	{0:1: 65535}	Value of Radio Network Temporary Identifier (UE-ID)	
	PRB allocation	U32	Binary	Each bit represents 4 PRBs (according to DL resource allocation type 0. For more information, refer to section 7.1.6.1 of *TS36.213: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (Release 10), V10.12.0* [3]). The leftmost bit represents the lowest resource block (group) index.	
	MCS	U8	{0:1:28}	The supported MCS value range is restricted to 0...28. MCS 29, 30 and 31 are not supported as there is no HARQ processing included.	

5.1.5.2.2 DCI_TX_CONFIG

Field		Type/ Length	Value range	Description	32-bit words
sub-messages of PHY_DL_TX_CONFIG_REQ					
	sub-message type (parameters set)	U8	**1**	Sub-message type defined for **1: DCI_TX_CONFIG**	

Field		Type/ Length	Value range	Description	32-bit words
	parameter set ID (parameter set version)	U8		Refer to section 3.6	
	parameter set body length	U24	**8**	Parameter set body length in bytes. Without the fields **sub-message type**, **parameter set ID**, and **parameter set body length**).	
	RNTI	U16	{0:1: 65535}	Value of Radio Network Temporary Identifier (UE-ID)	
	CCE Offset	U8	{0:2:14}	CCE offset value (1 CCE = 9 REG = 36 RE)	
	PRB allocation	U32	Binary	Each bit represents 4 PRBs (according to DL resource allocation type 0. For more information, refer to section 7.1.6.1 of *TS36.213: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (Release 10), V10.12.0* [3]. The leftmost bit represents the lowest resource block (group) index.	
	MCS	U8	{0:1:28}	The supported MCS value range is restricted to 0...28. MCS 29, 30 and 31 are not supported as there is no HARQ processing included.	
	TPC	U8		Not used	

5.1.5.3 PHY_DL_TX_PAYLOAD_REQ

This message is used by the eNodeB L2/MAC to provide MAC-PDU to eNodeB L1/PHY for PDSCH transmission. It is synchronized to DLSCH_TX_CONFIG through PDU index.

Field	Type/ Length	Value range	Description	32bit words
message type ID	U16	**0x4502**	Message type identifier	2xU32

Field	Type/ Length	Value range	Description	32bit words
reference ID	U16		Refer to section 3.4	
instance ID	U8		Refer to section 3.4	
message body length	U24		Message body length in bytes	
SFN	U16		Refer to section 5.1.4	2xU32
TTI index	U8		Refer to section 5.1.4	
number of sub-messages	U8		Number of sub-messages contained in the hierarchical message	
confirm mode	U8		Refer to section 5.1.4	
Reserved	U24		Refer to section 5.1.4	
For number of sub-messages (MAC PDUs) in PHY_DL_TX_PAYLOAD_REQ				
sub-message type (parameters set)	U8	**0**	Sub-message type defined for PHY_DL_TX_PAYLOAD_REQ 0: DLSCH_MAC_PDU	
parameter set ID (parameter set version)	U8		Refer to section 3.6	
parameter set body length	U24		Parameter set body length in bytes. Without the fields **sub-message type**, **parameter set ID**, and **parameter set body length**.	
PDU index	U8		Index to associate the MAC PDU in this sub-message with the corresponding transport channel configuration sub-message in the DL_TX_PAYLOAD_REQ o incremented which each transport channel configuration sub-message included in the DL_TX_CONFIG_REQ	

Field		Type/ Length	Value range	Description	32bit words
				○ reset to 0 with every new subframe or TTI	
	MAC PDU size	U24	Acc. to LTE TF table	MAC PDU size in bytes without padding bytes	
	MAC PDU	< MAC PDU size> x U8		Content and length depend on the content type defined by the parameter set. Content type "direct data": direct MAC PDU data provided by N = ceil(<MAC PDU size>/4) U32 words; if necessary, the least significant bytes of the last U32 words are zero padded	

5.1.5.4 PHY_ULSCH_RX_IND

This message is used by the eNodeB L1/PHY to inform the L2/MAC about received ULSCH transmissions and to provide the related decoding results. In case of CRC result failed the MAC PDU SIZE is zero and MAC PDU is omitted.

Field	Type/ Length	Value range	Description	32-bit words
message type ID	U16	**0x4401**	Message type identifier	2xU32
reference ID	U16		Refer to section 3.4	
instance ID	U8		Refer to section 3.4	
message body length	U24		Message body length in bytes	
SFN	U16		Refer to section 5.1.4	2xU32
TTI index	U8		Refer to section 5.1.4	
number of sub-messages	U8		Number of sub-messages contained in the hierarchical message	
confirm mode	U8		Refer to section 5.1.4	
Reserved	U24		Refer to section 5.1.4	
For number of sub-messages (MAC PDUs) in PHY_ULSCH_RX_IND				

	sub-message type (parameters set)	U8	**0**	Sub-message type defined for PHY_ULSCH_RX_IND 0: ULSCH_MAC_PDU	
	parameter set ID (parameter set version)	U8		Refer to section 3.6	
	parameter set body length	U24		Parameter set body length in bytes. Without the fields **sub-message type**, **parameter set ID**, and **parameter set body length**.	
	RNTI	U16	{0:1: 65535}	Value of Radio Network Temporary Identifier (UE-ID)	
	CRC Result	U8	{0;1}	For debug and statistics 0: fail 1: pass	
	MAC PDU size	U24	Acc. to LTE TF table	MAC PDU size (=TB size) in bytes (without padding bytes) = 0 for CRC Result == 0 (fail) > 0 for CRC Result == 1 (pass)	
	MAC PDU	< MAC PDU size> x U8		Content and length depend on the content type defined by the parameter set. Content type *direct data*: direct MAC PDU data provided by N = ceil(<MAC PDU size>/4) U32 words; if necessary, the least significant bytes of the last U32 words are zero padded.	

5.1.5.5 PHY_DLSCH_RX_IND

This message is used by the UE L1/PHY to inform the UE L2/MAC about received DLSCH transmissions and to provide the related decoding results. In case of CRC result failed the MAC PDU SIZE is zero and MAC PDU is omitted.

Field	Type/ Length	Value range	Description	32-bit words
message type ID	U16	**0x4801**	Message type identifier	2xU32
reference ID	U16		Refer to section 3.4	
instance ID	U8		Refer to section 3.4	
message body length	U24		Message body length in bytes	
SFN	U16		Refer to section 5.1.4	2xU32
TTI index	U8		Refer to section 5.1.4	
number of sub-messages	U8		Number of sub-messages contained in the hierarchical message	
confirm mode	U8		Refer to section 5.1.4	
Reserved	U24		Refer to section 5.1.4	
For number of sub-messages (in PHY_DLSCH_RX_IND)				
sub-message type (parameters set)	U8	**0**	Sub-message type defined for PHY_DLSCH_RX_IND 0: DLSCH_MAC_PDU	
parameter set ID (parameter set version)	U8		Refer to section 3.6	
parameter set body length	U24		Parameter set body length in bytes. Without the fields **sub-message type**, **parameter set ID**, and **parameter set body length**.	
RNTI	U16	{0:1: 65535}	Value of Radio Network Temporary Identifier (UE-ID)	
CRC Result	U8	{0;1}	For debug, statistics, and so on: 0: fail 1: pass	

Field		Type/ Length	Value range	Description	32-bit words
	MAC PDU size	U24	Acc. to LTE TF table	MAC PDU size (=TB size) in bytes (without padding bytes) = 0 for CRC Result == 0 (fail) > 0 for CRC Result == 1 (pass)	
	MAC PDU	< MAC PDU size> x U8		Actual MAC PDU data Content and length depend on the content type defined by the parameter set Content type *direct data*: direct MAC PDU data provided by N = ceil(<MAC PDU size>/4) U32 words; if necessary, the least significant bytes of the last U32 words are zero padded.	

5.1.5.6 PHY_UL_TX_PAYLOAD_REQ

This message is used by the UE L2/MAC to provide MAC-PDU to L1/PHY for PUSCH transmission.

Field	Type/ Length	Value range	Description	32-bit words
message type ID	U16	**0x4902**	Message type identifier	2xU32
reference ID	U16		Refer to section 3.4	
instance ID	U8		Refer to section 3.4	
message body length	U24		Message body length in bytes	
SFN	U16		Refer to section 5.1.4	2xU32
TTI index	U8		Refer to section 5.1.4	
number of sub-messages	U8		Number of sub-messages contained in the hierarchical message	
confirm mode	U8		Refer to section 5.1.4	
Reserved	U24		Refer to section 5.1.4	

Field		Type/ Length	Value range	Description	32-bit words
For number of sub-messages (in PHY_UL_TX_PAYLOAD_REQ)					
	sub-message type (parameters set)	U8	**0**	Sub-message type defined for PHY_UL_TX_PAYLOAD_REQ 0: ULSCH_TX_MAC_PDU	
	parameter set ID (parameter set version)	U8		Refer to section 3.6	
	parameter set body length	U24		Parameter set body length in bytes. Without the fields **sub-message type**, **parameter set ID**, and **parameter set body length**.	
	PDU index	U8		Index to associate the MAC PDU in this sub-message with the corresponding transport channel configuration sub-message in the UL_TX_CONFIG_REQ. • incremented which each transport channel configuration sub-message included in the UL_TX_CONFIG_REQ • reset to 0 with every new subframe or TTI	
	MAC PDU size	U24	Acc. to LTE TF table	MAC PDU size in bytes without padding bytes	
	MAC PDU	< MAC PDU size> x U8		Actual MAC PDU data Content and length depend on the content type defined by the parameter set. Content type *direct data*: direct MAC PDU data provided by N = ceil(<MAC PDU size>/4) U32 words; if necessary, the	

Field		Type/ Length	Value range	Description	32-bit words
				least significant bytes of the last U32 words are zero padded	

5.1.5.7 PHY_CNF

This message is used to indicate confirmations of any request message. The message should be sent to the instance that it was received from. It will be only generated if confirm mode is switched on.

Field		Type/ Length	Value range	Description	32-bit words
message type ID		U16	**0x4501**	message type identifier	2xU32
reference ID		U16		Filled with reference ID extracted from the incoming request for which the confirm is generated.	
instance ID		U8		Refer to section 3.4	
message body length		U24		Message body length in bytes	
SFN		U16		Refer to section 5.1.4	2xU32
TTI index		U8		Refer to section 5.1.4	
number of sub-messages		U8		Number of sub-messages contained in the hierarchical message	
confirm mode		U8		Refer to section 5.1.4	
Reserved		U24		Refer to section 5.1.4	
For number of sub-messages (in PHY_CNF)					
	sub-message type (parameters set)	U8	**0**	Sub-message type	
	parameter set ID (parameter set version)	U8		Refer to section 3.6	
	parameter set body length	U24	**3**	Parameter set body length in bytes. Without the fields **sub-message type**, **parameter set**	

Field		Type/ Length	Value range	Description	32-bit words
				ID, and **parameter set body length**.	
	Confirmation status	U8		Confirmation status (refer to section 0)	
	Source Message type ID	U16		Message ID for which CNF message is sent (refer to section 3.3)	

The PHY CNF supports the following confirmation states.

Value	Name	Description
0	SUCCESS	Request accepted and sent to the subsequent module; no error occurred during handling of the request
1	UNKNOWN MESSAGE	The primitive (message type) is unknown/undefined
2	MESSAGE NOT SUPPORTED	The primitive (message type) is defined, but not supported at the specific interface
3	UNKNOWN PARAMETER (SET)	One or more of the received parameter sets (or parameters) are not part of the message
4	MISSING PARAMETER (SET)	One or more of the (mandatory) parameter sets (or parameters) of the received message are missing
5	PARAMETER (SET) REPETITION	One or more of the received parameter sets (or parameters) have been set more than once
6	RANGE VIOLATION	One or more parameters are out of the supported range
7	STATE VIOLATION	The received request is not supported in the current state of operation. For example, this may indicate one of the following: • any violation of the allowed message sequence protocols • specific configurations which are not allowed due to previously configured settings / configurations
8	TIMEOUT	An expected message has not been received within the specific time period after a previous event (message)

Value	Name	Description
9	CONFIG PAYLOAD MISMATCH	The config part of a request does not fit to the payload part (for example, for mismatch of lengths)
10	LENGTH MISMATCH	The content of any **message body length** or **parameter set body length** indicator field does not match to the actual message body length or parameter set body length
11	INPUT BUFFER FULL	Request rejected since the input buffer for incoming requests is full. The subsequent module needs some time to execute the previously stored requests.
12	INTERNAL ERROR	Internal error
13	Instance ID Mismatch	The local instance ID is not like the decoded instance ID from API module

5.2 Component and Namespace Layout

All project elements can be organized in components, and all VIs can be unequivocally identified by namespaces. A component can be assigned easily to one or more targets in SystemDesigner. A component must not include any element that is not suited for the target the component is assigned to.

5.2.1 Component Layout

Because each element of the component should be openable on the target the component is assigned to, NI recommends that you separate module implementations into one component for FPGA and one for Host code. If Host and FPGA share some resources like typedef, create a third component to be used by both the Host and FPGA components.

5.2.2 Namespacing

Each element of a component receives a namespace which consists of the root namespace of the component and an inner component namespace. Note that the folder hierarchy of the component maps 1:1 to the inner namespace as shown in Figure 47. The namespace layout is independent of the component layout; two components can share the same root namespace as shown in Figure 48.

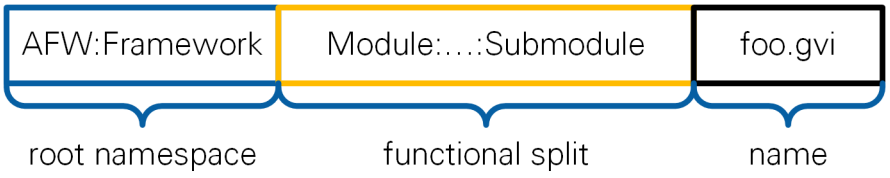

Figure 47 Namespace Hierarchy

All application frameworks components have `AFW` as the first level of their namespace. `AFW` is followed by the name of the framework as the second hierarchy

level. Elements that are shared between several frameworks have `AFW:Common` as their root namespace. The component target, such as FPGA or Host, is never part of the namespace. The inner namespace corresponds to the functional split inside the component.

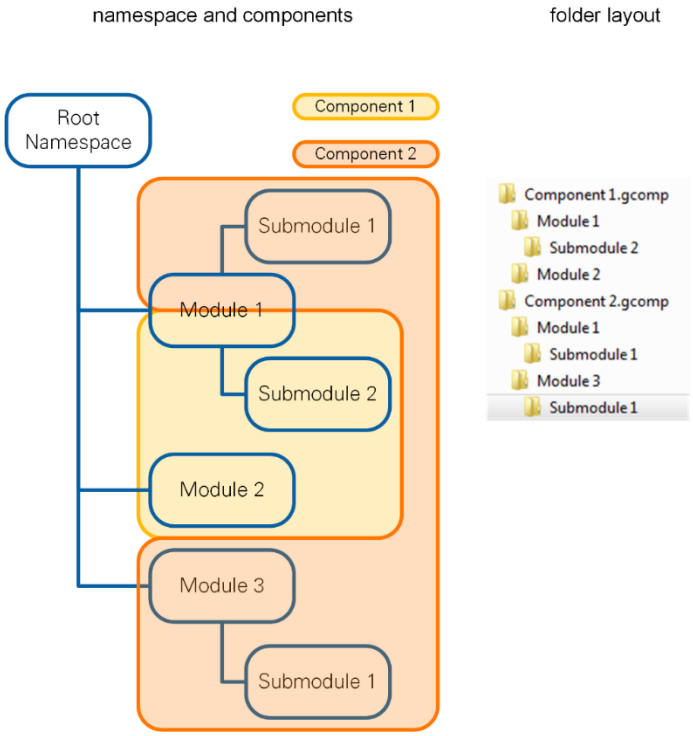

Figure 48 Connection between Namespaces, Components and Folder Layout on Disk

5.2.3 Extension of the Framework

To prevent conflicts with current and future versions of the framework, choose a root namespace that does not start with AFW.

To ease integration with future versions of the frameworks, keep the framework components untouched and put new VIs into separate components.

6 Abbreviations

Abbreviation	Meaning
3GPP	3rd Generation Partnership Project
ACK	Acknowledgement
ADC	Analog digital converter
BLER	Block error rate
CCE	Control channel elements
CFI	Control format indicator
CFO	Carrier frequency offset
CINR	Carrier to interference noise ratio
CP	Cyclic prefix
CRC	Cyclic redundancy check
CRS	Cell-specific reference signals
CSI	Channel state information
DAC	Digital analog converter

Abbreviation	Meaning
DCI	Downlink control information
DFT	Discrete Fourier transform
DL	Downlink
DMRS	Demodulation reference signals
eNodeB	Base station
FAM	Frontend adapter module (RF module)
FDD	Frame structure type 1
FFO	Fractional frequency offset
FIR	Finite impulse response
HARQ	Hybrid automatic repeat request
IFFT	Inverse fast Fourier transform
IFO	Integer frequency offset
LS	Least squares
LTE	Long-term evolution
LUT	Look-up table
MAC	Medium access control layer
MBSFN	Multimedia broadcast multicast service single frequency network
MCS	Modulation and coding scheme
MIMO	Multiple-input, multiple-output
NACK	Negative acknowledgment
OFDM	Orthogonal frequency-division multiplexing
OFDMA	Orthogonal frequency-division multiplexing
PBCH	Physical broadcast channel
PCFICH	Physical hybrid-ARQ indicator channel
PDCCH	PHY downlink control channel
PDSCH	Physical downlink shared channel
PHICH	Physical control format indicator channel
PHY	Physical layer
PMCH	Physical multicast channel
PRACH	Physical random access channel
PRB	Physical resource block
PSS	Primary synchronization signal
PUCCH	Physical uplink control channel
PUSCH	Physical uplink shared channel
QPSK	Quadrature phase-shift keying
RBG	Resource block groups
RE	Resource elements
REG	Resource element groups
RNTI	Radio network temporary identifier
RT	Real-time
RX	Receiver
SAP	Service access point
SC-FDMA	Single-carrier frequency-division multiplexing access
SDR	Software defined radio
SFN	System frame number
SINR	Signal-to-interference-plus-noise ratio

Abbreviation	Meaning
SISO	Single input, single output
SRS	Sounding reference symbols
SSS	Secondary synchronization signal
TB	Transport block
TDD	Frame structure type 2
TX	Transmitter
UDP	User datagram protocol
UE	User equipment
UERS	UE-specific reference signals
UL	Uplink
UL-SCH	Uplink shared channel
UpPTS	Uplink pilot time slot

7 Bibliography

[1] 3GPP, "TS36.211: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 10), V10.7.0," 2013-02.

[2] 3GPP, "TS36.212: Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding (Release 10), V10.8.0," 2013-06.

[3] 3GPP, "TS36.213: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (Release 10), V10.12.0," 2014-03.

377713D-01 Nov 2019