MANUAL

LabVIEW Communications MIMO Application Framework 19.5

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

Table of Contents

NATIONAL INSTRUMENTS

1 Introduction

Massive MIMO (multiple-input multiple-output) [1], deploying hundreds of antenna elements at the base station to serve multiple users on the same time-frequency resource, is a key component of future 5G cellular networks. Yet there are many open research questions to be answered to prove the viability and gains of Massive MIMO in practice.

LabVIEW Communications MIMO Application Framework software enables researchers to prototype Massive MIMO testbeds rapidly using LabVIEW Communications System Design Suite and state-of-the-art USRP Software Defined Radio Reconfigurable Devices and USRP Software Defined Radio Stand-Alone Devices (USRP devices). The application framework is fully open, modular, and modifiable. It allows researchers to set up their Massive MIMO testbed quickly and focus on modifying selected aspects of the system.

The application framework implements important features of an orthogonal frequency division multiplexing (OFDM) physical layer, with third generation partnership project (3GPP) long-term evolution (LTE) time-division duplex (TDD)-like specifications. The physical layer processes 20 MHz of signal bandwidth in real-time using FPGAs. It includes, variable modulation schemes, MIMO equalization, channel reciprocity based linear precoding, reciprocity calibration, synchronization functionality, and other features. The application framework supports up to 128 antennas at the base station with up to 12 single-antenna mobile stations or a multi-antenna mobile station with up to 12 antennas. That is, researchers can prototype a multiuser MIMO system with up to 12 single-antenna mobile stations, or a single user MIMO system including one mobile station with up to 12 antennas.

Researchers can modify the application framework to investigate their own (distributed) antenna arrays, different MIMO array signal processing techniques, or parameter estimation strategies under real world conditions, to name only a few examples.

This manual introduces the underlying system specifications in chapter 2. Chapter 3 covers signal processing algorithms. The mapping of system functionality to hardware modules is presented in chapter 4. Chapters 5 through 8 present implementation details of the base station, the single-antenna mobile station, the multi-antenna mobile station and the host controller. Over-the-air experimentation examples are summarized in chapter 9. Chapter 10 concludes this manual.

It is recommended to review the *MIMO Prototyping System Getting Started Guide* [2] for a brief overview over the MIMO Application Framework software and hardware architecture before studying the more detailed presentation in this Manual.

2 Overview

This chapter provides an overview of the application framework main features in section 2.1. Physical layer parameters are covered in section 2.2. Section 2.3 introduces medium access functionality.

2.1 System Features

The application framework offers a ready-to-run design for MIMO prototyping. It comprises a fully-functional physical layer for bi-directional transmission as well as basic medium access control layer elements. The main features are highlighted below.

- Multi-user MIMO transmission between one base station (BS) with up to 128 antennas and up to 12 single-antenna mobile stations (MS).
- Single-user MIMO transmission between one base station with up to 128 antennas and one mobile station with up to 12 antennas.
- Bi-directional, OFDM-based TDD transmission using 20 MHz signal bandwidth.
- Fully re-configurable LTE-like radio frame structure.
- Number of mobile stations or the number of spatial layers per multi-antenna mobile station can be changed at runtime. The system adapts automatically.
- Scalable number of antennas (multi-antenna mobile station: between 2 and 12; base station: between 2 and 128). Interfaces and configuration adapt automatically.
- FPGA based real time signal processing such as modulation, over-the-air synchronization, MIMO equalization and MIMO precoding.
- Linear multi-user MIMO precoding and equalization for up to 128 x 12 antenna systems. Options: Minimum mean squared error (MMSE), zero-forcing (ZF), and maximum-ratio combining (MRC). Multi-user MIMO precoding (at the base station) is channel reciprocity based.
- Automatic channel reciprocity calibration for all radio units at the base station.
- Automatic gain control at the base station (BS) and mobile station (MS).
- Variable modulation schemes ranging from 4-quadrature amplitude modulation (QAM) to 256-QAM.
- Un-coded transmission.
- Basic medium access control (MAC) functionality supports packet-based user data transmission in downlink (DL) and uplink (UL) to enable data streaming applications, such as video transmission.

2.2 Physical Layer

2.2.1 General Description

The application framework implements parts of an LTE-like physical layer in TDD mode. Uplink and downlink are based on OFDM using 20 MHz channel bandwidth.

Each OFDM symbol carries a single signal type or channel as described below.

Primary Synchronization Signal (PSS) (OFDM Symbol Type: Sync)
The PSS is compliant with the LTE standard [3]. The PSS sequence generation is implemented statically for Cell-ID 0.

The PSS is transmitted once per radio frame, that is, with a periodicity of 10 ms, instead of a periodicity of 5 ms as defined in the LTE standard. This modification is used to detect the radio frame start uniquely, in absence of a secondary synchronization sequence (SSS).

The PSS can be mapped to arbitrary DL OFDM symbol indices within a radio frame.

MS-Specific Reference Signals (OFDM Symbol Type: DL Pilot, UL Pilot)
The same length-1200 quadrature phase-shift keying (QPSK) sequence is used to derive uplink and downlink pilots. This sequence is derived from a randomly generated length-300 QPSK sequence which is repeated four times. Pilot sequences which correspond to different spatial layers (or mobile stations) use a length-100 subset of this sequence. This subset is derived as shown in Figure 2-1 for the example of spatial layers 1 and 7.
Pilot sequences corresponding to different spatial layers are transmitted in a frequency orthogonal fashion. A pilot corresponding to spatial layer 1 occupies the first subcarrier in each resource block (RB). A pilot corresponding to spatial layer 2 occupies the second subcarrier in each RB and so forth, as shown in Figure 2-1.

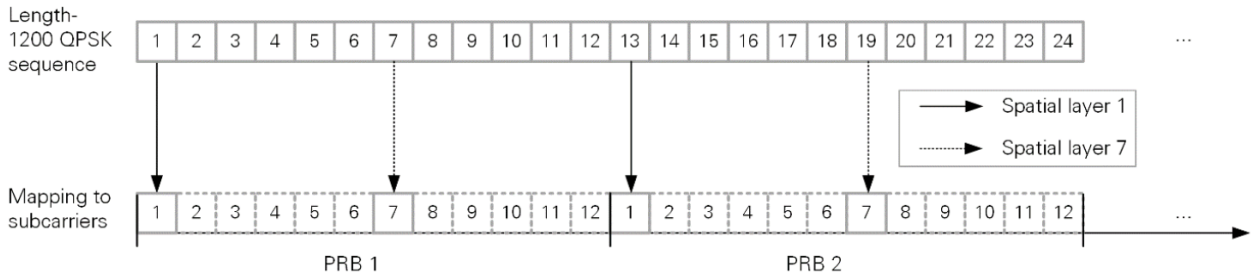

Figure 2-1. Selection of the Spatial Layer-Specific Pilot Sequence and Mapping to Subcarriers

Note that DL pilots are precoded before transmission, identical to data symbols. UL pilots, on the other hand, are transmitted without precoding.

Physical Downlink Shared Channel (PDSCH) (OFDM Symbol Type: DL Data)
Downlink payload data is transmitted over the PDSCH without forward error correction coding. Uncoded transport blocks of exact length are provided to the physical layer as described in section 2.3.3. PDSCH processing comprises:

- Scrambling using a polynomial in accordance to the LTE specifications [3].
  Note that the scrambler is initialized per OFDM symbol, based on the OFDM symbol number and the spatial layer.
- Modulation in accordance to the LTE specifications [3].

Physical Uplink Shared Channel (PUSCH) (OFDM Symbol Type: UL Data)
The PUSCH is symmetric to the PDSCH.

Guard (OFDM Symbol Type: Guard)
One OFDM symbol including cyclic prefix (CP) where nothing is transmitted in uplink or downlink direction. The guard is typically used for TDD switching. However, this is not required in the application framework as it is designed for transmission ranges that can be covered by the CP duration.

2.2.2 Numerology

Table 2-1 summarizes the main system parameters.

Table 2-1. MIMO Application Framework Numerology

Parameter	Value
Center Frequency[1] [GHz]	0.05 - 6
Channel bandwidth [MHz]	20
Sampling rate [MHz]	30.72
Subcarrier spacing [kHz]	15
Fast Fourier transform (FT) Size	2048
Number of used subcarriers	1200
Number of RBs	100
OFDM symbol duration [us]	66.67
OFDM symbol length [number of samples]	2048
CP duration [us] (Normal CP mode)	5.21 (first OFDM symbol in a slot), 4.69 (remaining OFDM symbols in a slot)
CP length [number of samples] (Normal CP mode)	160 (first OFDM symbol in a slot), 144 (remaining OFDM symbols in a slot)
OFDM symbols per slot	7
Slot duration [ms]	0.5
Modulation format	4-QAM, 16-QAM, 64-QAM, 256-QAM
Physical layer uncoded peak rate [Mbit/s] (12 layers, 256-QAM)	1612.8
Number of BS antennas	2, 4, 6, 8, 10, …, 128
Number of MS antennas	1, 2, 4, 6, 8, 10,12
Number of spatial layers	1, 2, 3, 4, ...,12

[1] Dependent on hardware specifications.

2.2.3 Radio Frame Format

The application framework uses a standard 3GPP LTE radio frame structure as shown in Figure 2-2. Radio frames of 10 ms duration are sub-divided into 10 subframes of 1 ms duration each. Each subframe contains two slots of 0.5 ms duration. A slot is further split into 7 OFDM symbols.

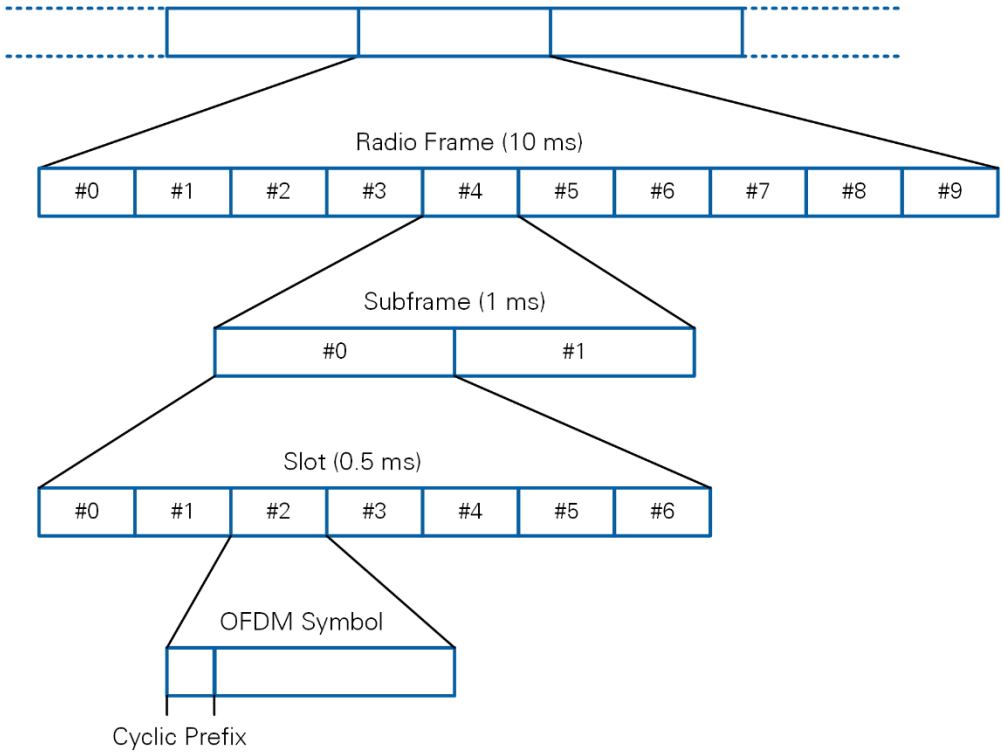

Figure 2-2. Radio Frame Structure

The application framework uses pre-configured radio frame schedules which are assigned on initialization time and remain static during run time. A radio frame schedule assigns downlink and uplink OFDM symbol types (see section 2.2.1) to each OFDM symbol within a radio frame. The system uses this schedule repetitively during an experiment.

The radio frame schedule can be chosen almost arbitrarily by the user, if the following conditions are satisfied:

- A radio frame must contain exactly one sync symbol at an arbitrary position. The sync symbol must be preceded by a DL type symbol (DL pilot, DL data or Guard) such that TDD switching is carried out in the CP of the symbol which precedes the sync symbol but not within the CP of the sync symbol.
- A radio frame must contain at least two consecutive downlink pilot symbols.
  - These two consecutive DL pilot symbols must be scheduled before any non-consecutive DL pilot symbols.
  - No data symbols may be transmitted before the two consecutive DL pilot symbols.
  - An arbitrary number of DL pilot symbols can be scheduled.

- A radio frame must contain at least one uplink pilot symbol.
- A new channel estimate is obtained whenever a pilot symbol is received. The data symbol which follows this pilot symbol is equalized using this new channel estimate already. The channel estimate is held constant until the next pilot symbol is received. That is, the channel estimate becomes more and more outdated for data symbols with larger distance to the pilot symbol received last. Higher order constellation symbols should be transmitted immediately after a pilot symbol.
- There is no need to use guard symbols when switching between uplink and downlink. Uplink and downlink can be switched on a per OFDM symbol basis.
- The radio frame schedule is not signaled from the base station to the mobile stations. Thus, the same schedule needs to be configured at base station and mobile stations.

Note that the frequency division duplex (FDD) frame timing according to [3] is used. That is, the uplink and downlink OFDM symbol timing is calibrated to start at the exact same point in time at the physical antenna port.

2.3 Medium Access Control

The application framework supports a subset of data path-related MAC functionality. This section presents an overview of the implemented features and details of the respective packet structures. Related physical layer processing is introduced for completeness as well.

2.3.1 Overview

The application framework supports the transmission of packet-based payload, for instance for user datagram protocol (UDP)-based video transmission, throughput measurements, or for pseudo noise (PN) sequence transmission. Figure 2-3 shows an overview how two payload packets are passed through the data path at transmitter and receiver.

The following steps are executed:

1) Data Transfer Protocol (DTP) Packet Processing
   A custom data transfer protocol is introduced to encapsulate payload packets, determine their position within a received byte stream and test for correct reception of a packet.
   a) Transmitter: Encapsulation of payload packets (UDP or PN data) into DTP packets
   b) Receiver: Extraction of DTP packets from Transport blocks, extraction of payload packets from DTP packets and generation of receive indications
2) Transport Block Processing
   Transport block processing further encapsulates parts of DTP packets. It allows to check for error free reception of transport blocks by means of a cyclic redundancy check (CRC).
   a) Transmitter: Encapsulation of DTP packets into transport blocks, modulation and mapping to OFDM symbols

b) Receiver: Equalization, hard bit decision and formatting into transport blocks.

Note that DTP packets and transport blocks each comprise individual CRC information. Received packets are only forwarded if both CRC checks indicate an error free reception.

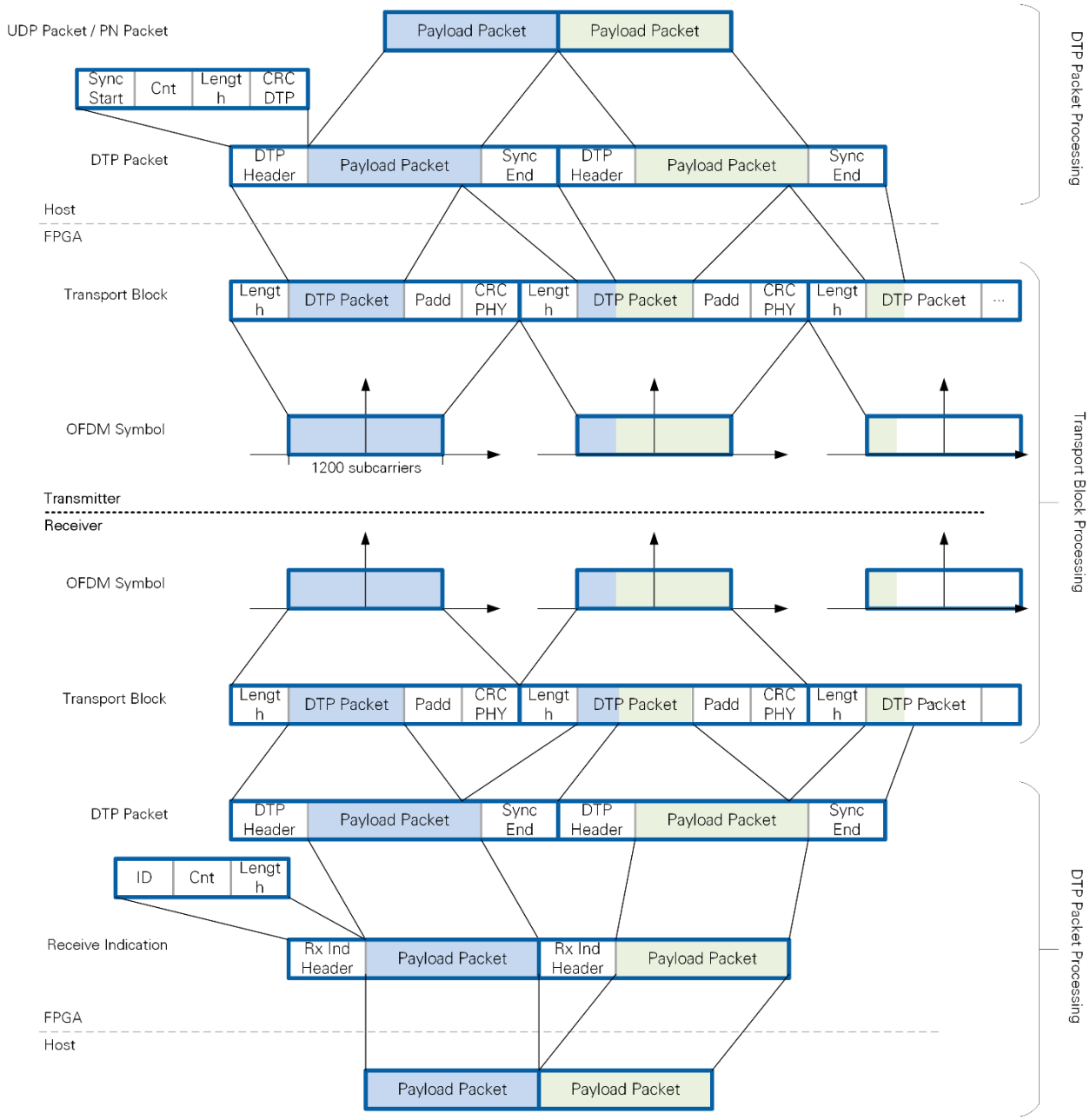

Figure 2-3. Data Packets Passing through the MIMO Application Framework Transmit and Receive Path

2.3.2 Data Transfer Protocol Processing

The data transfer protocol encapsulates UDP or PN payload packets, containing an arbitrary number of bytes, into DTP packets, by adding header and tail information. Note that the number of bytes per payload packet plus any additional headers is only restricted by the FPGA implementation to be less than 16,385 bytes. The header and tail information

allows to reconstruct the payload packets at the receiver. The DTP packet structure is shown in Figure 2-3 and explained in Table 2-2.

Table 2-2. DTP Packet Structure

Field Name	Length in Bytes	Description
Sync Start	4	Indicates the start of a DTP packet, constant value 0x2EA1DA7A
Cnt	2	User definable value to signal the packet number and detect missing packets at the receiver
Length	2	Length of the payload packet measured in bytes
CRC DTP	4	CRC computed from the DTP header
Payload Packet	variable	Actual payload data of variable length, source is either UDP or a PN generator
Sync End	4	Indicates the end of a DTP packet, constant value 0x2EA1DA7B

Payload is encapsulated into DTP packets at the transmitter host. On the other hand, DTP packet processing mainly resides at the receiver FPGA for computational complexity reasons.

At the transmitter, DTP packets are transferred to the FPGA through DMA first-in first-out memory buffers (FIFOs). A DTP packet can get segmented and transmitted using multiple transport blocks as described in the following section. Likewise, a transport block can contain multiple DTP packets. The Sync Start and Sync End fields in each DTP packet allow to extract DTP packets from transport blocks at the receiver. The example in Figure 2-3 shows how two payload packets are mapped to two DTP packets. The first DTP packet is mapped to the first two transport blocks. The second DTP packet is mapped to the second and third transport block.

Note that the CRC, which is part of the DTP header, is used to verify the correctness of the header at the receiver, in addition to evaluating the Sync Start and Sync End fields.

Payload packets are extracted from DTP packets at the receiver FPGA, based on DTP header information. Each payload packet is prepended a Receive Indication Header before it is passed to the host. Receive Indication Header and Payload Packet form a receive indication. Table 2-3 summarizes the structure of the Receive Indication.

Table 2-3. Structure of the Receive Indication

Field Name	Length in Bytes	Description
MS-ID	2	Signals the data source/sink corresponding to this payload packet.
Cnt	2	User-definable value to signal the packet number and detect missing packets at the receiver (from DTP header)
Length	2	Length of the payload packet measured in bytes

Field Name	Length in Bytes	Description
Payload Packet	variable	Actual payload data of variable length

2.3.3 Transport Block Processing

The purpose of the transport block processing is to generate transport blocks from DTP packets. Each transport block is required to have the exact length that can be mapped to a single OFDM symbol. This length depends on the modulation scheme and the static number of subcarriers per OFDM symbol. That is, each OFDM symbol carries a single transport block. The transport block length derives from

$$transport\ block\ length\ [bytes] = \frac{number\ of\ subcarriers \cdot bits\ per\ modulation\ symbol}{8}.$$

Table 2-4 summarizes the transport block length for different modulation schemes for a static number of 1,200 occupied subcarriers.

Table 2-4. Transport Block Lengths for Different Modulation Schemes

Modulation	Bits per symbol	Packet length [bytes]
QPSK	2	300
16-QAM	4	600
64-QAM	6	900
256-QAM	8	1200

Transport blocks are assembled on the FPGA from DTP packets. It may happen that a DTP packet is not fully available on the FPGA at the time a transport block is generated. Likewise, a DTP packet might be too long to be mapped to a single transport block. In the former case, the part of the DTP packet which is already available is mapped into the transport block. Padding bytes are added to fill the transport block as shown in Figure 2-3. In the latter case, a fraction of the DTP packet is mapped into the transport block. No padding is added in this case. The Length field captures the number of DTP packet bytes included in the transport block. Note that multiple (partial) DTP packets may be mapped into a transport block, for instance a fraction of DTP packet i, followed by DTP packet $i + 1$ and the first part of DTP packet $i + 2$. Note also that padding bytes are chosen to be zero. The scrambling operation, which is one of the following steps in transmission chain, ensures a random bit pattern.

Table 2-5 summarizes the transport block structure.

Table 2-5. Transport Block Structure

Field Name	Length in Bytes	Description
Length	4	Number of DTP packet bytes transmitted in the current transport block
DTP Packet	variable	Part of a DTP packet or of multiple DTP packets
Padd	variable	Padding of bytes to complete a transport block.
CRC PHY	4	Cyclic redundancy code (CRC) computed from the length field and the DTP packet

The CRC PHY is used to verify whether the transport block has been received without errors. The polynomial definition of the CRC is chosen according to IEEE Standard 802.11-2012 Section 8.2.4.8 of *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications* [4].

2.3.4 Transport Block-to-Spatial Layer Mapping

The application framework supports the transmission of up to 12 spatial layers. Each mobile station can be assigned a subset of these spatial layers for uplink transmission and downlink reception. Likewise, the base station provides 12 data sources, that is, 12 UDP ports for transmission of UDP packets or 12 PN generators. Each data source is uniquely coupled to a mobile station, identified by a MS-ID. This subsection lists the rules how data sources are mapped to spatial layers and hence to mobile stations.

Each of the up to 12 data sources is assigned its own DTP and transport block processing, independent of all other data sources, as described in sections 2.3.2 and 2.3.3. The resulting transport blocks are mapped to spatial layers as shown in Figure 2-4. Note that a complete transport block is mapped to a certain layer before a new transport block is mapped to another layer. That is, a transport block is not split between multiple spatial layers.

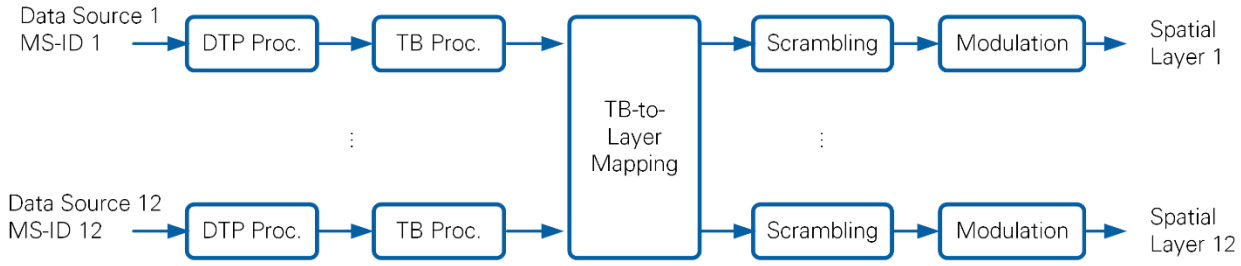

Figure 2-4. Transport Block-to-Layer Mapping at the Base Station

The TB-to-Layer Mapping follows the set of rules listed below.

1. A single spatial layer must be assigned a single data source. For instance, spatial layer 3 can be assigned data source 12, but no other data sources.
2. A data source can be mapped to multiple layers. For instance, data source 12 can be mapped to spatial layers 3, 4, 5.

3. A mobile station must not be assigned multiple data sources but only a single data source. For instance, if a mobile station is assigned spatial layers 3,4,5, then, the data source assigned to this mobile station must be mapped to these spatial layers and no other spatial layers.

This mapping operation can be represented in terms of a size 12x12 mapping matrix. Feasible examples are shown below in (2.1). Columns enumerate spatial layers. Rows enumerate data sources or MS-IDs, respectively. Rule 1) tells that there can only be a single x per column, rule 2) tells that there can be 0 to 12 xs per column.

$$\begin{bmatrix} x & x & x & x & x & x & x & x & x & x & x & x \\ - & - & - & - & - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - & - & - & - & - \end{bmatrix}, \begin{bmatrix} x & - & - & - & - & - & - & - & - & - & - & - \\ - & x & - & - & - & - & - & - & - & - & - & - \\ - & - & x & - & - & - & - & - & - & - & - & - \\ - & - & - & x & - & - & - & - & - & - & - & - \\ - & - & - & - & x & - & - & - & - & - & - & - \\ - & - & - & - & - & x & - & - & - & - & - & - \\ - & - & - & - & - & - & x & - & - & - & - & - \\ - & - & - & - & - & - & - & x & - & - & - & - \\ - & - & - & - & - & - & - & - & x & - & - & - \\ - & - & - & - & - & - & - & - & - & x & - & - \\ - & - & - & - & - & - & - & - & - & - & x & - \\ - & - & - & - & - & - & - & - & - & - & - & x \end{bmatrix}, \begin{bmatrix} x & x & x & - & - & - & - & - & - & - & - & - \\ - & - & - & x & x & x & - & - & - & - & - & - \\ - & - & - & - & - & - & x & x & x & - & - & - \\ - & - & - & - & - & - & - & - & - & x & x & x \\ - & - & - & - & - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - & - & - & - & - \\ - & - & - & - & - & - & - & - & - & - & - & - \end{bmatrix}. \quad (2.1)$$

rows = data sources, columns = layers

The left-hand side matrix represents the case that all layers obtain transport blocks from data source 1. That is, there is a single mobile station which is assigned 12 layers. The other data sources remain unused.
The center matrix represents the case that layer i obtains transport blocks from data source i. That is, there are 12 mobile stations in the example, each assigned 1 layer.
The right-hand side matrix represents the case that layers 1, 2, and 3 obtain transport blocks from source 1, layers 4, 5, and 6 from source 2, and so forth.

Note that the mapping also permits no data transmission on a certain spatial layer.

2.3.5 Downlink and Uplink Scheduling

Users can define a radio frame schedule as presented in section 2.2.3. This radio frame schedule defines the OFDM symbol type for each OFDM symbol in a radio frame. The transmission follows this radio frame schedule repetitively.

Each spatial layer uses the same radio frame schedule. A different modulation scheme can be assigned to UL and DL data symbols per layer. Mobile stations can be assigned spatial layers on run time. However, all spatial layers use the same time-frequency resources.

3 Signal Processing

This chapter provides an overview over the physical layer transceiver signal processing block diagrams of the base station, the single antenna mobile station and the multi-antenna mobile station in section 3.1. Detailed descriptions of main algorithms follow in subsequent sections. Note that this chapter does not cover algorithm implementation specific details. These are covered in chapters 5, 6, 7, and 8.

3.1 Signal Processing Overview

This section presents an overview of the transmit and receive signal processing at the base station, the single- and the multi-antenna mobile station, respectively. Details about the individual signal processing operations are provided in following sections.

3.1.1 Base Station

Figure 3-1 shows the base station signal processing block diagram including downlink transmitter and uplink receiver. The grey boxes indicate hardware units which implement the respective signal processing functionality, for later reference.

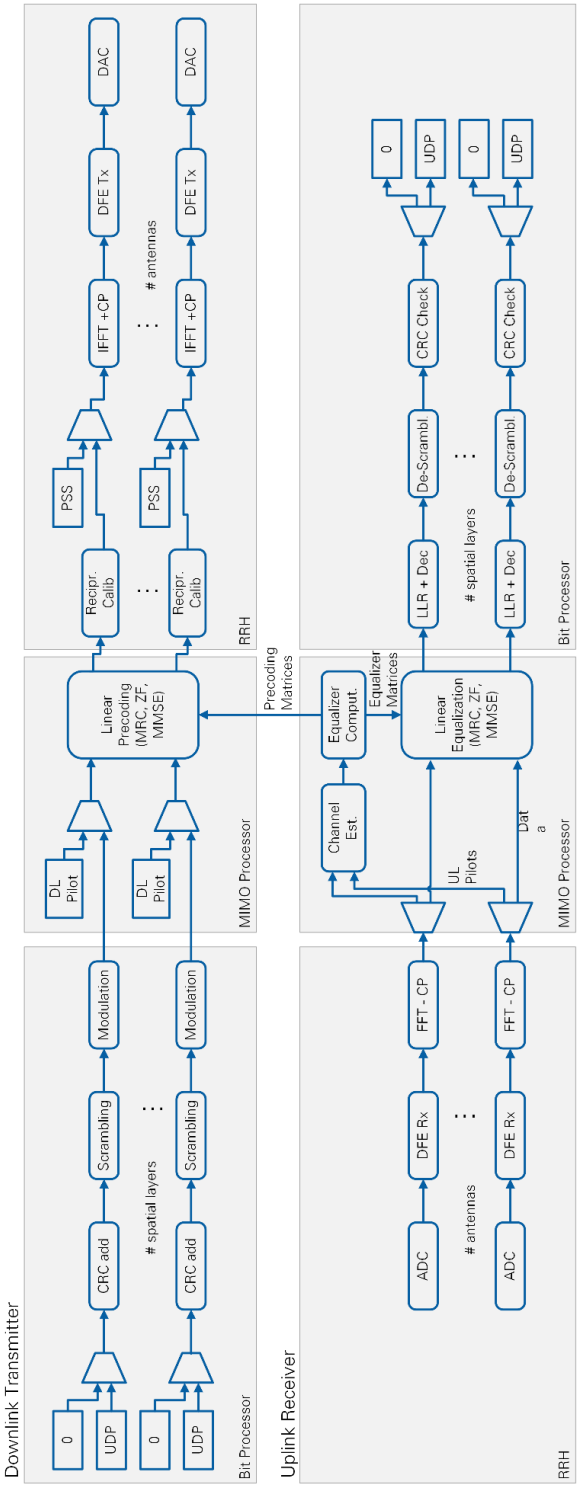

Figure 3-1. Base Station Signal Processing Block Diagram

Transmitter

- The base station transmitter generates signals for a configurable number of spatial layers. Data is transmitted uncoded, that is, without forward error correction.

- Per spatial layer, either payload data is received at a UDP data interface, or zeros are transmitted. Payload data is appended zeros or it is segmented as needed.
- The **CRC add block** appends a frame check sequence (FCS) to the payload. In addition, it prepends a header containing the length of the UDP payload.
- The **Scrambler block** scrambles the binary input sequence. The scrambler ensures the transmission of a zero-mean random binary sequence, even if the input sequence comprised only zero values. See section 3.10.
- The **Modulation block** maps groups of bits to M-QAM signals. {4, 16, 64, 256} QAM are supported.
- Downlink pilot signals are inserted, depending on the frame schedule. Note that the pilot position within a physical resource block depends on the layer. Pilots assigned to different layers are frequency orthogonal.
- Spatial layers are mapped to antennas in frequency domain by means of the **Linear Precoding block**. This block will be discussed in more detail in section 3.5. Note that precoding is applied to downlink pilot and data signals.
- Reciprocity calibration coefficients are applied in frequency domain per antenna by the **Reciprocity Calibration block** to compensate for the impact of non-reciprocal transmit and receive radio transceiver modules at the base station and to enable reciprocity based precoding. Reciprocity calibration will be discussed in more detail in section 3.6.
- The Primary Synchronization Sequence is inserted, depending on the frame schedule. Note that this sequence is not precoded and is transmitted from all antennas.
- The signal is converted to time domain per antenna by means of a size-2048 FFT. A cyclic prefix is prepended.
- The transmit digital frontend block **DFE Tx** resamples the signal from 30.72 MHz to the native sampling frequency of the digital-to-analog converters.

Receiver

- The signal at the output of each analog-to-digital converter is fed into the receive digital frontend block **DFE Rx**. This block implements automatic DC offset correction and resampling from the native sampling frequency of the ADC to 30.72 MHz.
- The mobile stations are responsible for correcting carrier frequency offsets in their transmit signals and for adjusting their transmit timing such that the uplink signal arrives synchronous to the downlink frame timing. Hence, no further synchronization functionality is implemented at the base station receiver.
- The cyclic prefix is discarded and the signal is converted to frequency domain by means of a size-2048 FFT (**FFT-CP block**).
- Uplink pilot signals are extracted from the frequency domain receive signal and fed into a channel estimation block

- The **Channel Estimation** block computes channel transfer functions for all transmit-receive antenna combinations. That is, up to 128 x 12 = 1,536 different channel transfer functions are computed in total. Channel estimation is covered in more detail in section 3.3.
- All channel estimates are provided to the **Equalizer Computation block**. This block computes MIMO equalization matrices to be applied at each subcarrier by the **Linear Equalization block** to obtain estimates for the transmitted symbols. MIMO equalization matrices can be computed according to the maximum ratio combining, zero forcing, or minimum mean square error criterion. MIMO Equalization is covered in more detail in section 3.4.
- The **LLR + Dec block** computes log likelihood ratios for all bits comprised in an equalized symbol, followed by a binary hard decision, denoted hard bits.
- The **De-Scrambling block** applies an XOR operation with the scrambling sequence to the hard bits.
- The **CRC Check block** computes the CRC check sum based on the de-scrambled bits. The result is used to detect any hard decision errors in a sequence of hard bits.
- Header information is used to separate UDP payload and transmitted zeros.

3.1.2 Single-Antenna Mobile Station

Figure 3-2 shows the single-antenna mobile station signal processing block diagram including uplink transmitter and downlink receiver. Many signal processing blocks perform the same operation as compared to the base station. The differences will be highlighted in this section.

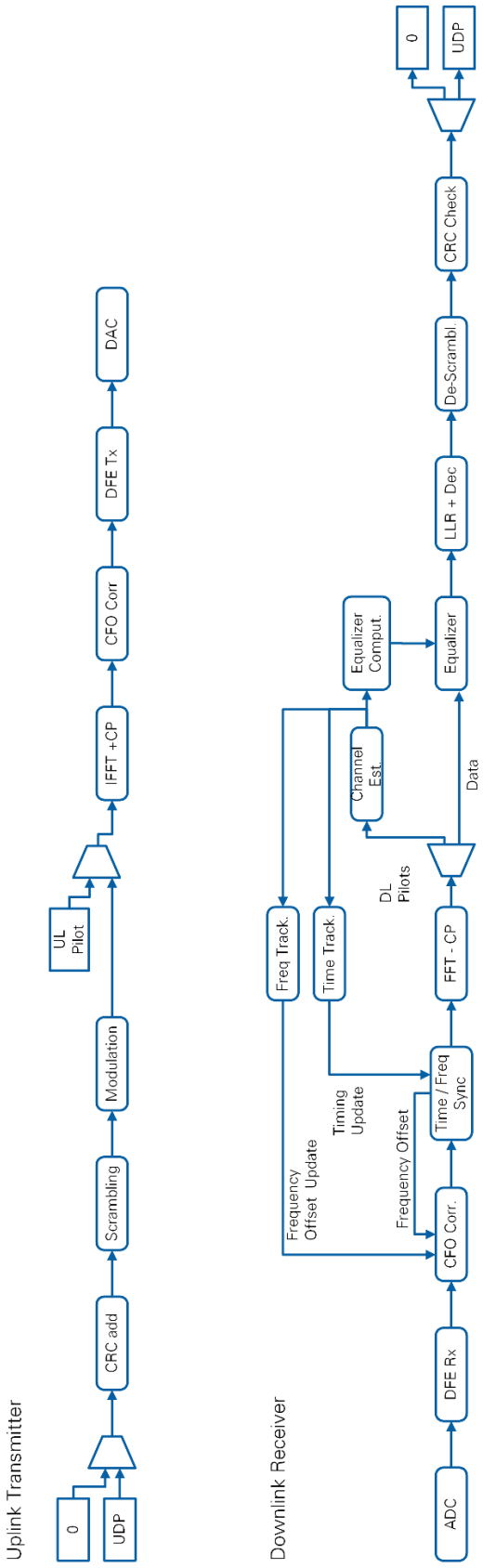

Figure 3-2. Single-Antenna Mobile Station Signal Processing Block Diagram

Uplink Transmitter

- A single spatial layer is transmitted from a single antenna.
- The transmitter is identical to the base station transmitter up to and including the **Modulation block**.
- Uplink pilots are transmitted instead of downlink pilots.
- No Linear Precoding or Reciprocity Calibration is applied. No primary synchronization sequence is inserted before the IFFT.
- A **CFO correction block** corrects the for the frequency offset measured in the downlink such that the uplink signals from several mobile stations are received frequency synchronous at the base station.
- The transmit digital frontend block **DFE Tx** is identical to the base station transmitter.

Downlink Receiver

- The signal at the output of each analog-to-digital converter is fed into the receive digital frontend block **DFE Rx** similar to the base station
- In addition to the receive processing at the base station, the mobile station receiver implements time and frequency synchronization and tracking. This functionality is explained in detail in section 3.7.
- The **CFO Correction block** compensates for the carrier frequency offset estimated by following **frequency synchronization** and **frequency tracking blocks**.
- The **Time/Frequency Synchronization block** computes the start of a radio frame based on cross-correlation with the primary synchronization sequence. The start of an OFDM symbol is computed based on the autocorrelation of the cyclic prefix with the end of an OFDM symbol. These initial estimates are subsequently refined by a **Time Tracking block**. The autocorrelation of the CP and the end of an OFDM symbol is also used to compute an initial estimate for the carrier frequency offset, which is further refined by the frequency tracking block.
- The time and frequency synchronized signal is converted to frequency domain by means of a size-2048 FFT after discarding the cyclic prefix.
- Downlink pilots are used to estimate the channel transfer function by the **Channel Estimation block**. The linear part of the channel phase (versus frequency) is used to estimate remaining timing synchronization inaccuracies by the **Time Tracking block**. The phase difference of two subsequent channel estimates is used to estimate the residual carrier frequency offset by the **Frequency Tracking block**.
- The channel estimate is used to compute channel equalizer coefficients (**Equalizer Computation block**), followed by the actual equalization (**Equalizer block**) similar to the base station.
- The remaining signal processing blocks perform identical operations as compared to the base station.

3.1.3 Multi-Antenna Mobile Station

The multi-antenna mobile station (see Figure 3-3) extends the single antenna mobile station by adding the capability to transmit and receive multiple spatial streams using multiple antennas. The multi-antenna processing heavily borrows from the base station signal processing as explained in this section.

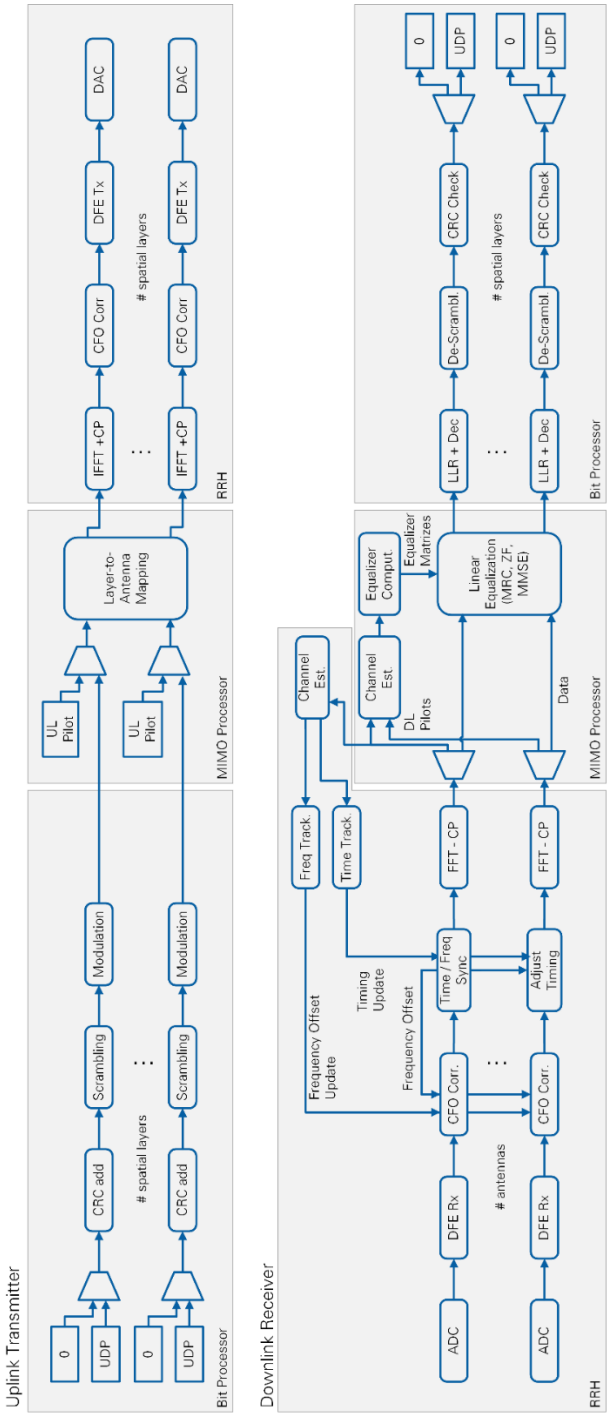

Figure 3-3. Multi-Antenna Mobile Station Signal Processing Block Diagrams

Uplink Transmitter

- The processing is identical to the single antenna mobile station except for the transmission of multiple layers. There are as many spatial layers as active transmit antennas, up to the number of transmit antennas supported by the mobile station hardware. The number of active layers can be selected by the user.
- The **Layer-to-Antenna Mapping block** maps one layer to one transmit antenna. A single layer is transmitted from antenna 1, two layers are transmitted from antennas 1 and 2, and so on. There is no precoding on the uplink transmitter

Downlink Receiver

- The receive processing up to the equalization part is identical to the single antenna mobile station, except for synchronization
- A single receive antenna is used for time and frequency synchronization as well as tracking, similar to the single antenna mobile station. The synchronization results are shared with other antennas and applied there.
- The same **Channel Estimation**, **Equalizer Computation**, **Linear Equalization blocks** are used as compared to the base station receiver.
- The remaining processing is similar to the single-antenna mobile station.

3.2 Synchronous Frequency Domain Transmission Model

This section introduces the mathematical transmission model in frequency domain at a single subcarrier. The model assumes perfect time and frequency synchronization. Algorithms presented in following sections are derived based on this model.

3.2.1 Notation

The following notation will be used throughout this chapter. Any deviations will be highlighted.

- $\boldsymbol{H}$ channel matrix frequency domain, uplink: $\boldsymbol{H} \in C^{M \times S}$, downlink: $\boldsymbol{H} \in C^{K \times M}$
- $\boldsymbol{W}$ Equalizer matrix $\in C^{S \times M,K}$
- $\boldsymbol{F}$ Precoding matrix $\in C^{M \times S}$
- $\boldsymbol{B}$ Reciprocity calibration matrix $\in C^{M \times M}$
- $\boldsymbol{x}$ Transmit signal vector $\in C^{S \times 1}$
- $\boldsymbol{y}$ Receive signal vector $\in C^{M,K \times 1}$
- $\boldsymbol{v}$ White Gaussian noise vector $\in C^{M,K \times 1}$
- $\widetilde{\boldsymbol{H}}$ Effective channel including precoding, $\widetilde{\boldsymbol{H}} = \boldsymbol{HF}$
- σ^2 Inverse signal-to-noise ratio (SNR) in linear domain
- N_{SC} # subcarriers,
- i subcarrier index
- N_{FFT} FFT size
- M # Base station antennas
- K # Mobile stations or mobile station antennas
- S # Spatial layers

- m, k index of the base station and mobile station antennas respectively
- $(\cdot)_{UL}, (\cdot)_{DL}$ indicates up- and downlink respectively, where needed
- $\hat{\boldsymbol{a}}$ denotes an estimate of $\boldsymbol{a}$

Boldface letters denote vectors and matrices. Normal letters denote scalar values. The m-th column or row of a matric $\boldsymbol{A}$ is denoted $\boldsymbol{A}_{:,m}$ and $\boldsymbol{A}_{m,:}$, respectively. a_m is the m-th element in vector $\boldsymbol{a}$.

3.2.2 Downlink Transmission Model

The downlink signal is transmitted in a precoded fashion from the base station. The signal received at all mobile station antennas at a single subcarrier can be modelled as

$$\boldsymbol{y} = \boldsymbol{HFx} + \boldsymbol{v}. \tag{3.1}$$

The effective wireless channel including precoding is

$$\widetilde{\boldsymbol{H}} = \boldsymbol{HF}. \tag{3.2}$$

In case of multiple **single-antenna mobile stations**, there are as many receive antennas as there are spatial layers, that is, $K = S$. $\boldsymbol{y}$ and $\boldsymbol{x}$ have the same number of elements in this case. The precoding operation has the goal to direct the useful signal x_1 to y_1, x_2 to y_2 and so forth, while suppressing interference from other layers.

The signal received at the k-th mobile station antenna is

$$y_k = \widetilde{H}_{k,k} x_k + \sum_{j \neq k} \widetilde{H}_{k,j} x_j + v_k \,. \tag{3.3}$$

The summation term represents residual spatial interference. This term equals zero if interference is completely suppressed by the precoding. Otherwise it can be viewed as additional noise source which can be combined with the Gaussian noise v_k.

In case of a **multi-antenna mobile station,** there can be more receive antennas than spatial layers. The dimension of $\widetilde{\boldsymbol{H}}$ is $K \times S$ in this case.

3.2.3 Uplink Transmission Model

The uplink signal is transmitted without precoding. The signal received at all base station antennas at a single subcarrier can be modelled as

$$\boldsymbol{y} = \boldsymbol{Hx} + \boldsymbol{v}. \tag{3.4}$$

Note that the relation $\boldsymbol{H}_{UL} = (\boldsymbol{H}_{DL})^T$ is assumed to hold (channel reciprocity), as a TDD system is considered.

The dimension of $\boldsymbol{H}_{UL}$ is $M \times K$ in case of multiple single-antenna mobile stations. In case of a multi-antenna mobile station, the dimension is $M \times S$ where $S \leq K$, as transmit antennas are turned off in case fewer spatial layers than antennas shall be transmitted (see section 3.9).

3.3 Channel Estimation

Channel estimation is implemented in the frequency domain. It relies on frequency orthogonal pilots transmitted in uplink and downlink, respectively. Note that uplink pilots are designed to be frequency orthogonal per antenna, while downlink pilots are frequency orthogonal per spatial layer. Downlink pilots are transmitted with precoding, similar to the actual data. That is, a pilot corresponding to one of the spatial layers is transmitted from all base station antennas in the downlink.

Consider the signal received at a pilot position p (subcarrier occupied by a pilot symbol) in the uplink at one base station receive antenna. It originates from a single mobile station antenna

$$y_p = x_p H_p + v_p. \tag{3.5}$$

A similar equation can be formulated for the downlink based on (3.3)

$$y_p = x_p \widetilde{H}_p + v_p.$$

Note that the spatial interference term is zero in this case as pilots transmitted on different spatial layers are frequency orthogonal. Note also, that $\widetilde{H}_p$ is the effective channel including precoding, which is to be estimated.

Using the uplink transmission model, the channel coefficient H_p at a pilot subcarrier is estimated as shown below

$$\widehat{H}_p = y_p \frac{x_p^*}{|x_p|^2} = H_p + \frac{x_p^*}{|x_p|^2} v_p. \tag{3.6}$$

Channel coefficients at non-pilot subcarrier position are obtained through interpolation as shown in Figure 3-4.

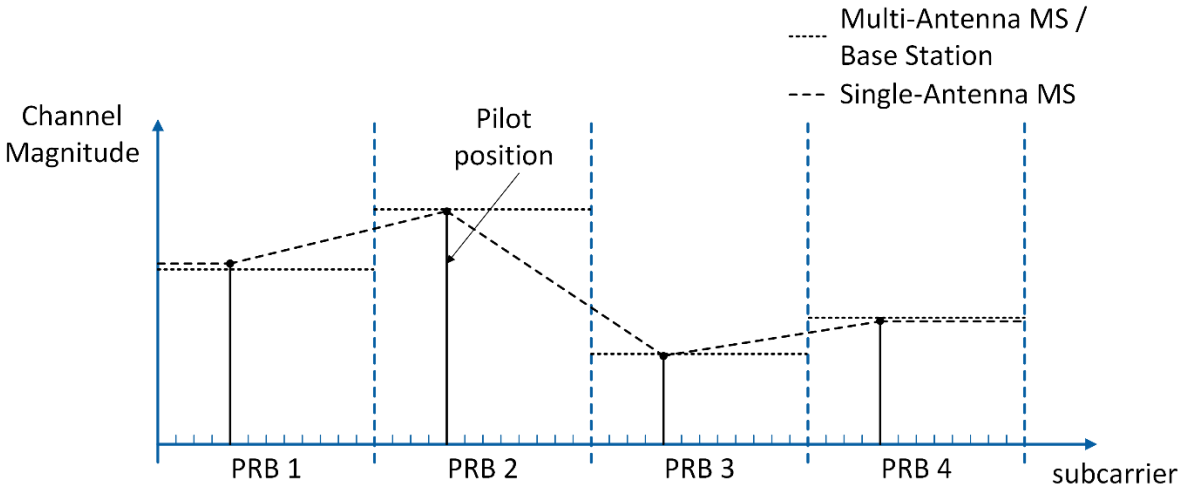

Figure 3-4. Channel Estimation and Interpolation Principle

Two different interpolation mechanisms are employed depending on the device type, as follows:

1) Multi-antenna MS/BS
   The channel coefficient measured at the pilot position is used throughout the respective RB (zero order hold interpolation).
2) Single-antenna MS
   The channel between two pilot symbols is obtained from linear interpolation. At the band edges, the channel is kept constant at the value measured at the closest pilot symbol (zero order hold interpolation) as shown in Figure 3-4.

The different choice in interpolation mechanisms is dictated by the mapping of algorithms to hardware modules which is explained later. In case of the multi-antenna MS or the BS, different PRBs are processed at different FPGAs (channel estimation and MIMO processing).

The channel transfer function obtained by this operation is used for equalization purposes starting from the OFDM symbol following the pilot OFDM symbol. The channel estimate is kept constant until the next OFDM pilot symbol is received, depending on the frame schedule.

Constraints:

- The coherence bandwidth of the channel should be larger than an RB such that the block constant channel estimate per RB is accurate.
- The coherence time of the channel should be larger than the distance between pilot OFDM symbols in time domain, such that the block constant channel estimate in time direction is applicable.
- Imperfect timing synchronization leads to a linear phase rotation over frequency. This linear phase drift is not accounted for by the zero-order hold interpolation used by the multi-antenna MS and the BS. Thus, timing synchronization in uplink and downlink need to be sufficiently accurate.

3.4 SISO and MIMO Equalization

3.4.1 Downlink SISO Equalization

This equalization method applies to the single antenna mobile station receiver in the downlink. A channel estimate is available at each subcarrier through linear interpolation as explained in the previous section. These channel estimates are used to compute an equalization coefficient per subcarrier.

Consider the data signal received at a certain subcarrier at a certain single-antenna mobile station in frequency domain.

$$y = \tilde{H}x + v. \tag{3.7}$$

This equation does not consider the effect of any residual spatial interference (see (3.3)). This interference, if present, is included in v and not considered separately. An estimate for $\widetilde{H}$ is available as described in the channel estimation section.

Assuming perfect channel knowledge, an estimate of the transmitted symbol is obtained as follows:

$$\hat{x} = Wy, \qquad W = \frac{\widetilde{H}^*}{\left|\widetilde{H}\right|^2}. \tag{3.8}$$

$\widehat{\widetilde{H}}$ is an estimated version of $\widetilde{H}$, which in practice is used to compute the equalization weight W.

3.4.2 Linear MIMO Equalization for the Up- and Downlink

The uplink transmission equation (3.4) is used to derive the equalizer matrix $\boldsymbol{W}$ in this section. However, this operation is applicable to the downlink receiver of the multi-antenna mobile station as well. Note that a single equalizer matrix is computed per RB. It is used to equalize all signals received at all subcarriers in this RB. Note also that the channel estimate is kept constant within one RB as described in the channel estimation section.

An estimate for the transmitted signal vector is obtained by multiplying the received signal vector with an equalizer matrix $\boldsymbol{W}$ as follows

$$\hat{\boldsymbol{x}} = \boldsymbol{W}\boldsymbol{y}. \tag{3.9}$$

$\boldsymbol{W}$ can be selected according to the following criteria

$$\boldsymbol{W}_{MRC} = \boldsymbol{C}\boldsymbol{H}^H. \tag{3.10}$$

$$\boldsymbol{W}_{ZF} = (\boldsymbol{H}^H\boldsymbol{H})^{-1}\boldsymbol{H}^H \tag{3.11}$$

$$\boldsymbol{W}_{MMSE} = (\boldsymbol{H}^H\boldsymbol{H} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{H}^H \tag{3.12}$$

(3.10) represents maximum ratio MRC, (3.11) represents ZF equalization, and (3.12) represents MMSE equalization. $\sigma^2 = \sigma_v^2/\sigma_x^2$ in (3.12) represents the inverse SNR in linear domain. Note that $\boldsymbol{W}_{MRC}$ is computed based on normalized channel columns which ensure an unbiased estimate. This normalization operation is represented by the diagonal matrix $\boldsymbol{C}$ of dimension $K \times K$ at the base station**.** The k-th diagonal element derives from the column norm of $\boldsymbol{H}$ as shown below

$$C_{k,k} = \left(\sum_m \left|H_{m,k}\right|^2\right)^{-1}. \tag{3.13}$$

Note that the estimated channel is used to compute the equalization matrices. Note also that MMSE equalization matrix leads to a biased estimate. The bias is $\sqrt{tr(\boldsymbol{W}\boldsymbol{W}^H)}$.

Constraints:

- The coherence bandwidth of the channel should be much larger than an RB so that using the same constant equalization matrix in an RB is accurate.
- Imperfect timing synchronization leads to a linear phase rotation over frequency. This linear phase drift is not compensated by the single equalization matrix per RB. A residual phase rotation can remain after equalization.

3.4.3 Linear MIMO Equalization Using the QR Decomposition

Computing the equalization matrix $\boldsymbol{W}$ can be computationally simplified by means of a QR decomposition. The equivalent representation of the equalizer matrix in terms of QR decomposition is derived in this section for later reference.

Consider the MMSE extension $\boldsymbol{B}$ of the MIMO channel matrix $\boldsymbol{B} = \begin{bmatrix}\boldsymbol{H}\\ \sigma\boldsymbol{I}\end{bmatrix}$ [5]. The term in the inverse (3.12) can be represented as follows:

$$\boldsymbol{B}^H\boldsymbol{B} = [\boldsymbol{H}^H \ \ \sigma\boldsymbol{I}]\begin{bmatrix}\boldsymbol{H}\\ \sigma\boldsymbol{I}\end{bmatrix} = \boldsymbol{H}^H\boldsymbol{H} + \ \sigma^2\boldsymbol{I}. \tag{3.14}$$

Now consider the QR decomposition of $\boldsymbol{B}$:

$$\boldsymbol{B} = \boldsymbol{Q}\boldsymbol{R} = \begin{bmatrix}\boldsymbol{H}\\ \sigma\boldsymbol{I}\end{bmatrix} = \begin{bmatrix}\boldsymbol{Q}_1\\ \boldsymbol{Q}_2\end{bmatrix}\boldsymbol{R}. \tag{3.15}$$

$\boldsymbol{R}$ is an upper triangular matrix with a dimension of $K \times K$, $\boldsymbol{Q}_1$ has a dimension of $M \times K$, and $\boldsymbol{Q}_2$ has a dimension of $K \times K$. The following identities hold:

$$\boldsymbol{R}^{-1} = \frac{\boldsymbol{Q}_2}{\sigma}, \quad \boldsymbol{H} = \boldsymbol{Q}_1\boldsymbol{R}. \tag{3.16}$$

The MMSE equalizer can now be represented in terms of QR decomposition as follows

$$\boldsymbol{W}_{MMSE} = (\boldsymbol{B}^H\boldsymbol{B})^{-1}\boldsymbol{H}^H = \boldsymbol{R}^{-1}\boldsymbol{Q}_1^H = \frac{1}{\sigma}\boldsymbol{Q}_2\boldsymbol{Q}_1^H. \tag{3.17}$$

This alternative representation helps to avoid costly matrix inversion operations [6]. The $\boldsymbol{Q}$ matrix can be implemented using the modified Gram-Smith algorithm as presented in section 5.2.3.5.

3.5 Reciprocity Based Precoding

The (non-normalized) linear precoding matrix $\boldsymbol{F}'$ can be computed in a similar fashion as compared to the equalizer matrix. The MRC, ZF and MMSE variants are shown below

$$\boldsymbol{F}'_{MRC} = \boldsymbol{H}_{DL}^H. \tag{3.18}$$

$$\boldsymbol{F}'_{ZF} = \boldsymbol{H}_{DL}^H(\boldsymbol{H}_{DL}\boldsymbol{H}_{DL}^H)^{-1} \tag{3.19}$$

$$\boldsymbol{F}'_{MMSE} = \boldsymbol{H}_{DL}^H(\boldsymbol{H}_{DL}\boldsymbol{H}_{DL}^H + \sigma^2\boldsymbol{I})^{-1} \tag{3.20}$$

Note that these precoding matrices do not yet guarantee that any transmit power constraint is met. This is denoted by $\boldsymbol{F'}$. Normalization that yields $\boldsymbol{F}$ is explained further below. The MMSE precoding matrix assumes that no receiver side equalization is implemented, that is, $\boldsymbol{W} = \boldsymbol{I}$.

In TDD systems, with perfect calibration of transmit and receive radios and with a sufficiently long channel coherence time, it can be assumed that uplink and downlink channel are identical, that is, reciprocal, as shown below

$$\boldsymbol{H}_{DL} = \boldsymbol{H}_{UL}^{T}. \tag{3.21}$$

The MMSE precoding matrix can hence be represented in terms of uplink channel matrix as follows

$$\boldsymbol{F'}_{MMSE} = ((\boldsymbol{H}_{UL}^{H}\boldsymbol{H}_{UL} + \sigma^{2}\boldsymbol{I})^{-1}\boldsymbol{H}_{UL}^{H})^{T} = \boldsymbol{W}_{MMSE}^{T}. \tag{3.22}$$

That is, the downlink precoding matrix at the base station can be computed by transposing the uplink MIMO equalization matrix at the base station. The same can be shown for MRC and ZF based precoding. Using the estimated uplink channel coefficients alleviates the need to obtain a channel estimate in the downlink and to convey this estimate to the base stations, for example, using channel feedback.

A row in $\boldsymbol{F'}$ maps all spatial layers to a single transmit antenna. The transmit power at the m-th transmit antenna is $\sigma_x^2 \sum_s \left|F'_{m,s}\right|^2$. In order not to exceed unit transmit power, we normalize the precoding matrix as follows

$$\boldsymbol{F} = \beta \boldsymbol{F'}. \tag{3.23}$$

The normalization factor is chosen according to

$$\beta = \left(\max_m \sigma_x^2 \sum_s \left|F'_{m,s}\right|^2 \right)^{-1/2} \tag{3.24}$$

By normalization, non-linear distortions at the transmitter can be avoided.

Constraints:

- A single MIMO equalization matrix $\boldsymbol{W}$ is computed per RB. Hence, a single precoding matrix is computed per RB as well. Spatial interference is best suppressed if the channel is constant throughout one RB.
- The base station does not time-synchronize to the uplink signal. That is, the channel measured in the uplink will be subject to a linear phase rotation in frequency domain if the uplink signal does not arrive fully accurate aligned to the base station timing. The measured uplink channel is used to compute the precoding matrix which is applied to downlink pilots as well. The precoded downlink pilots contain the inverse linear phase rotation as compared to the uplink channel. This phase rotation is measured as part of the mobile station channel

estimation. It can impact the time tracking capabilities of the mobile station as the mobile station overestimates timing inaccuracies.

3.6 Reciprocity Calibration

In section 3.5 it was argued that the downlink precoding matrix $\boldsymbol{F}$ can be derived from the uplink MIMO channel equalization matrix $\boldsymbol{W}$, if uplink and downlink channels were reciprocal. This assumption is often not valid in practice, as a matter of variations in analog components. Reciprocity calibration can alleviate this issue as presented in this section.

3.6.1 The Need for Reciprocity Calibration

Reduced to a single arbitrary BS-MS antenna pair, indexed $\{m, k\}$ as shown in Figure 3-5, channel reciprocity means that the wireless channel response $H_{k\to m}^{UL}$ seen in the uplink is equal to channel response $H_{m\to k}^{DL}$ of the reverse wireless channel seen in the downlink

$$H_{m\to k}^{DL} = H_{k\to m}^{UL}. \tag{3.25}$$

In this case, it is sufficient to estimate the uplink channel responses $H_{k\to m}^{UL}$ between all mobile station antennas $1 \dots K$ and all base station antennas $1 \dots M$ to characterize the wireless MIMO channel in uplink and in downlink. Note that $H_{k\to m}^{UL}$ and $H_{m\to k}^{DL}$ each denote a frequency domain channel coefficient at an arbitrary subcarrier.

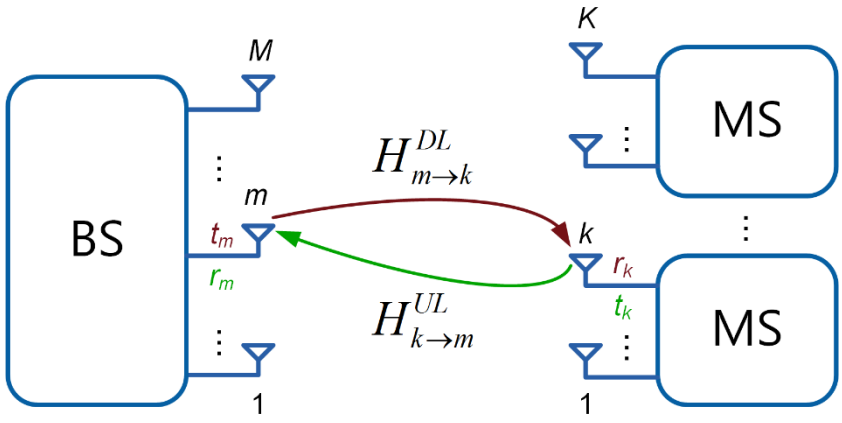

Figure 3-5. Multi-user MIMO System Model

In practical wireless systems, perfect reciprocity cannot be achieved even if the wireless part of the channel is reciprocal. The characteristics of analog radio transceiver components, which can be viewed as part of the channel, vary across different devices. Hence, the baseband receivers in the base station and the mobile stations measure the effective channel responses

$$\tilde{H}_{k\to m}^{UL} = t_k \cdot H_{k\to m}^{UL} \cdot r_m \tag{3.26}$$

$$\tilde{H}_{m\to k}^{DL} = t_m \cdot H_{m\to k}^{DL} \cdot r_k. \tag{3.27}$$

These effective channel responses include the transmit responses t_k, t_m and the receive responses r_m, r_k of the respective analog transceiver front ends connected to the BS antenna m and MS antenna k, in addition to the wireless channel $H_{m\to k}^{DL}$ and $H_{k\to m}^{UL}$. Due to analog component variations and dynamic effects caused by clocking structures, such as clock dividers, multipliers, and phase-locked loops (PLLs), it is practically impossible to realize analog front ends with identical reciprocal transmit and receive responses. Typically, random phase and magnitude differences are observed between the analog front end responses, which might change with every reset of the respective radio hardware. Therefore, the reciprocity condition does not hold for the effective channel responses, that is

$$\widetilde{H}_{m\to k}^{DL} \neq \widetilde{H}_{k\to m}^{UL}. \tag{3.28}$$

However, as shown in *Argos: Practical Many-Antenna Base Stations* [7], after a so-called reciprocity calibration at the base station, it is still possible to derive an estimate for the effective downlink channel response $\widetilde{H}_{m\to k}^{DL}$ from the estimated effective uplink channel response $\widetilde{H}_{m\to k}^{UL}$ if the physical wireless channel is reciprocal.

3.6.2 Reciprocity Calibration Fundamentals

As given in equations (3.26) and (3.27), the effective channel response $\widetilde{H}$ between any two transceivers can be expressed as the product of the transmit front end response t, the physical wireless channel responses h, and the receive front end response r.

A reciprocity calibration factor $b_{m\to k}$ can be defined as quotient between the channel responses of the effective forward channel and the effective backward channel.

$$b_{m\to k} = \frac{\widetilde{H}_{m\to k}^{DL}}{\widetilde{H}_{k\to m}^{UL}} = \frac{t_m \cdot H_{m\to k}^{DL} \cdot r_k}{t_k \cdot H_{k\to m}^{UL} \cdot r_m} \tag{3.29}$$

As long as the reciprocity condition (3.25) is valid for the wireless propagation channel between base station antenna m and mobile station antenna k, the reciprocity calibration coefficient $b_{m\to k}$ reduces to

$$b_{m\to k} = \frac{t_m \cdot H_{m\to k}^{DL} \cdot r_k}{t_k \cdot H_{k\to m}^{UL} \cdot r_k} = \frac{t_m \cdot r_k}{t_k \cdot r_m} = \frac{t_m}{r_m} \cdot \frac{r_k}{t_k} = b_{k\to m}^{-1} \tag{3.30}$$

and the corresponding effective downlink channel response can be calculated from the effective uplink channel response as follows

$$\widetilde{H}_{m\to k}^{DL} = \widetilde{H}_{k\to m}^{UL} \cdot b_{m\to k} = \widetilde{H}_{k\to m}^{UL} \cdot \frac{t_m \cdot r_k}{r_m \cdot t_k}. \tag{3.31}$$

In this case, the reciprocity calibration factor only depends on the transmit and receive responses of the involved analog front ends. Of course, this also works in the reverse direction. Generally, we can write

$$\widetilde{H}_{m\to k} = \widetilde{H}_{k\to m} \cdot b_{m\to k} = \frac{\widetilde{H}_{k\to m}}{b_{k\to m}} \quad \Leftrightarrow \quad \widetilde{H}_{k\to m} = \widetilde{H}_{m\to k} \cdot b_{k\to m} = \frac{\widetilde{H}_{m\to k}}{b_{m\to k}} \tag{3.32}$$

Equations (3.31) and (3.32) are still not practical for a direct implementation in real systems since they would require a measurement of the reciprocity coefficients between every BS and MS antenna pair, requiring an unacceptable pilot and feedback overhead.

Argos: Practical Many-Antenna Base Stations [7] proposes a BS-internal relative reciprocity calibration scheme that relies on reciprocity measurements at the BS only. For this, the BS estimates the reciprocity coefficients $b_{m\to s}$ between the front end of one fixed BS reference antenna r and the front ends of all other BS antennas $m \neq s$

$$b_{m\to s} = \frac{t_m \cdot H_{m\to s} \cdot r_s}{t_s \cdot H_{s\to m} \cdot r_m}\,. \tag{3.33}$$

For this purpose, known pilots are transmitted over the air forth and back between the BS antenna pairs (m, s) during the calibration procedure. As long as the reciprocity condition $H_{m\to s} = H_{s\to m}$ is valid for the wireless channel between the different BS antennas, the corresponding reciprocity factors only depend on the transmit and receive responses of the involved analog BS front ends.

$$b_{m\to s} = \frac{t_m \cdot r_s}{t_s \cdot r_m} \tag{3.34}$$

Note that the reciprocity factors $b_{m\to s}$ measured at the base station are assumed to be approximately stable over a longer period since all base station front ends are clock-coupled. Practically, the length of this period depends also on different front end properties and potential impairments, for example. phase noise characteristics, phase drifts caused by thermal effects, and so on.

With the BS reciprocity factors $b_{m\to s}$ defined above, the reciprocity factor between base station antenna m and mobile station antenna k can be calculated as follows

$$b_{m\to k} = \frac{t_m \cdot r_k}{r_m \cdot t_k} = \frac{t_m \cdot t_s \cdot r_s \cdot r_s}{r_m \cdot t_s \cdot r_s \cdot t_k} = \frac{t_m \cdot r_s}{r_m \cdot t_s} \cdot \frac{t_s \cdot r_k}{r_s \cdot t_k} = b_{m\to s} \cdot b_{s\to k} \tag{3.35}$$

Finally, the effective downlink channel response between an arbitrary BS antenna transceiver m and the MS antenna transceiver k can be derived from the effective uplink channel response as follows

$$\widetilde{H}^{DL}_{m\to k} = \widetilde{H}^{UL}_{k\to m} \cdot b_{m\to s} \cdot b_{s\to k}. \tag{3.36}$$

Thus, with the BS reciprocity calibration factors $b_{m\to s}$ and estimates for $\widetilde{H}^{UL}_{k\to m}$ all downlink channel responses $\widetilde{H}^{DL}_{m\to k}$ can be calculated up to a common factor $b_{s\to k}$. This common factor does not change the precoding characteristics and can be measured and compensated at the mobile station [7]. That is, it is sufficient to compute the linear precoding matrix from the effective channel

$$\widetilde{H}'^{DL}_{m\to k} = \widetilde{H}^{UL}_{k\to m} \cdot b_{m\to s}. \tag{3.37}$$

Note that the reciprocity coefficient for $m = s$ cannot be measured in practice. Mathematically it reduces to $b_{m\to m} = 1$.

3.6.3 Multi-Hop Estimation of Reciprocity Calibration Coefficients

Obtaining the downlink channel coefficient $\widetilde{H}'^{DL}_{m\to k}$ from the corresponding uplink channel coefficient $\widetilde{H}^{UL}_{k\to m}$ requires knowledge of the base station-internal reciprocity calibration factor $b_{m\to s}$ ((3.37)). This coefficient can be determined by estimating the channel coefficients $\widetilde{H}_{m\to s}$ and $\widetilde{H}_{s\to m}$ (3.33). These channel estimates may be very noisy, depending on the antenna array properties and the attenuation between antennas (s, m), resulting in an inaccurate estimate for $b_{m\to s}$. This in turn can cause the linear downlink precoding operation to be inaccurate and result in significant residual spatial interference at the mobile stations.

This problem can be alleviated by combining multiple estimates of $b_{m\to s}$ as follows.

1) Define a set $\boldsymbol{S}$ of reference antennas chosen from the set of base station antennas $\boldsymbol{S} = \{s_1 \dots s_S\}$ with $S \le M$. Note that the former reference antenna s in the further notation will correspond to s_1.
2) Expand the definition of $b_{m\to s_1}$ in (3.34) as follows

$$b_{m\to s_1} = \frac{t_m \cdot r_{s_1}}{r_m \cdot t_{s_1}} = \frac{t_m \cdot r_{s_2}}{r_m \cdot t_{s_2}} \cdot \frac{t_{s_2} \cdot r_{s_1}}{r_{s_2} \cdot t_{s_1}} = b_{m\to s_2} \cdot b_{s_2\to s_1} \tag{3.38}$$

   From (3.38) the calibration coefficient between two antennas can be split into the product of two calibration coefficients. That is, $b_{m\to s_1}$ can be obtained from the calibration coefficients between antennas (m, s_2) and antennas (s_2, s_1). This decomposition can be interpreted as estimating $b_{m\to s_1}$ in two hops.
3) More general, using the set of reference antennas $\mathcal{S}$, a large number of hops can be defined as shown in Figure 3-6. More formally, we can capture the different options for multi-hops in terms of index sets $\mathcal{A}_n$, $1 \le n \le N$. In Figure 3-6, for example, the set of reference antennas has been chosen as $\mathcal{S} = \{s_1, s_2, s_3, s_4\}$. Using these reference antennas, four different index sets have been defined $\mathcal{A}_1 = \{(m, s_4), (s_4, s_3), (s_3, s_2), (s_2, s_1)\}$, $\mathcal{A}_2 = \{(m, s_3), (s_3, s_2), (s_2, s_1)\}$, $\mathcal{A}_3 = \{(m, s_2), (s_2, s_1)\}$, $\mathcal{A}_4 = \{(m, s_1)\}$. Each index set allows to estimate $b_{m\to s_1}$ using different paths through the antenna array as shown in Figure 3-6. Note that it is possible to define further index sets and further reference antennas for the example shown in Figure 3-6.
4) Formally, we define $b_{m\to s_1|\mathcal{A}_n}$, which is the calibration coefficient $b_{m\to s_1}$ represented by the product of calibration coefficients defined in the index set $\mathcal{A}_n$

$$b_{m\to s_1|\mathcal{A}_n} = \prod_{\forall (a,b)\in\mathcal{A}_n} b_{a\to b}. \tag{3.39}$$

5) Assume N different estimates $\hat{b}_{m\to s_1|\mathcal{A}_n}$ have been obtained. These estimates can be linearly combined to obtain a reliable estimate $\hat{b}_{m\to s_1}$ of the reciprocity calibration coefficient $b_{m\to s_1}$.

$$\hat{b}_{m\to s_1} = \sum_{n=1}^{N} w_n \cdot \hat{b}_{m\to s_1|\mathcal{A}_n} \qquad \text{for} \quad m \neq s_1 \tag{3.40}$$

The weights w_n are derived from the variance $\mathrm{Var}\{\hat{b}_{m\to s_1|\mathcal{A}_n}\}$ of the estimates as follows

$$w_n = \frac{\dfrac{1}{\mathrm{Var}\{\hat{b}_{m\to s_1|\mathcal{A}_n}\}}}{\sum_{n'=1}^{N} \dfrac{1}{\mathrm{Var}\{\hat{b}_{m\to s_1|\mathcal{A}_{n'}}\}}}. \tag{3.41}$$

That is, estimates with smaller variance contribute more to the combined estimate.

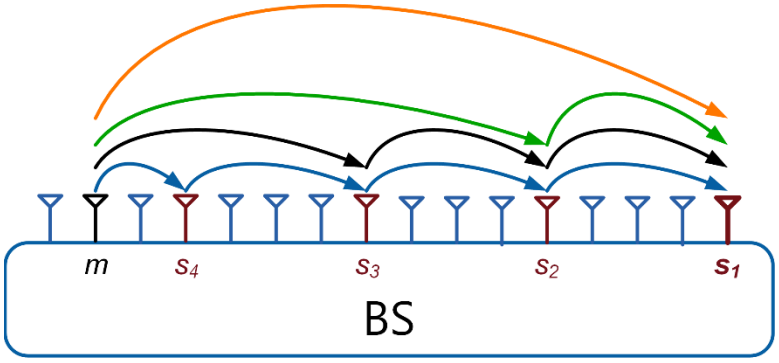

Figure 3-6. Multi-Hop Estimation of Reciprocity Coefficients

3.6.4 Reciprocity Calibration Algorithm

The actual reciprocity calibration algorithm, which uses the principles derived in section 3.6.3, is presented in this section.

The algorithm exploits the fact that reciprocity coefficients can be assumed to be constant within the used signal bandwidth which is small compared to the carrier frequency. That is, multiple estimates, obtained at different subcarriers can be further combined to suppress estimation noise.
Also, the algorithm reduces computational complexity by allowing only subsets of possible multi-hop paths, that is, a small number N of sets $\mathcal{A}_n$ is used to compute a certain calibration coefficient.

The algorithm executes as follows:

1) Defining the Reference Antenna Set:
Define a reference antenna index set $\mathcal{S}= \{s_1, s_2, \ldots, s_S\}$. Indices are chosen such that the

physical distance between reference antenna s_s and antenna s_1 increases with increasing index. Ensure a uniform distribution of the reference antennas across the base station antenna array.

2) Defining Multi-Hop Paths:
The number of possible multi-hop paths between antenna m and reference antenna s_1, that is, the number N of sets $\mathcal{A}_n$, is restricted to a single path per number of hops. This is achieved as follows (see Figure 3-6 for an example)

- The first hop of a dual hop set ends at antenna s_2, followed by one single hop $s_2 \rightarrow s_1$ to antenna s_1.
- The first hop of a triple hop ends at antenna s_3, followed by two single hops $s_3 \rightarrow s_2$, and $s_2 \rightarrow s_1$.
- The first hop of a quadruple hop ends at antenna s_4, followed by three single hops $s_4 \rightarrow s_3$, $s_3 \rightarrow s_2$, and $s_2 \rightarrow s_1$. And so on.

This rule ensures that there are as many sets $\mathcal{A}_n$ as reference antennas, that is, $S = N$. Note that these sets are identical for different BS antennas m, except for the first hop starting at antenna m.

3) Single-Hop Reciprocity Coefficient Measurement at all Subcarriers
Measure the reciprocity coefficients $\hat{b}_{m \rightarrow s_s}(i)$ between every BS antenna m and every reference antenna s_s for every sub-carrier i. This yields $M \cdot R$ (single-hop) measurements at each subcarrier. All further steps use this measurement data set.

1) Send calibration pilots $X_p(i)$ on every sub-carrier i forth and back between all BS antenna pairs (m, s_s), with $s_s \in \mathcal{S}$

$$Y_{m \rightarrow s_s}(i) = \widetilde{H}_{m \rightarrow s_s}(i) \cdot x_p(i) + v_{m \rightarrow s_s}(i) \quad \text{for} \quad m \neq s_s \tag{3.42}$$
$$Y_{s_s \rightarrow m}(i) = \widetilde{H}_{s_s \rightarrow m}(i) \cdot x_p(i) + v_{s_s \rightarrow m}(i) \quad \text{for} \quad s_s \neq m \tag{3.43}$$

   $Y(i)$ denotes the frequency-domain received signal and $v(i)$ for realization of random noise terms caused by TX and RX noise of the involved analog front ends.
2) Estimate the effective forward and backward channels between all BS antenna pairs (m, s_s), with $s_s \in \mathcal{S}$

$$\widehat{H}_{m \rightarrow s_s}(i) = Y_{m \rightarrow s_s}(i) \cdot x_p^*(i) = \widetilde{H}_{m \rightarrow s_s}(i) - v_{m \rightarrow s_s}(i) \cdot x_p^*(i) \quad \text{for} \quad m \neq s_s \tag{3.44}$$
$$\widehat{H}_{s_s \rightarrow m}(i) = Y_{s_s \rightarrow m}(i) \cdot x_p^*(i) = \widetilde{H}_{s_s \rightarrow m}(i) - v_{s_s \rightarrow m}(i) \cdot x_p^*(i) \quad \text{for} \quad s_s \neq m \tag{3.45}$$

   For simplicity pilots are assumed to have unit magnitude, that is, $|x_p(i)| = 1$.
3) Calculate the reciprocity calibration coefficient for all BS antenna pairs (m, s_s), with $s_s \in \mathcal{S}$ and sub-carrier i

$$\hat{b}_{m \rightarrow s_s}(i) = \frac{\widehat{H}_{m \rightarrow s_s}(i)}{\widehat{H}_{s_s \rightarrow m}(i)} \approx b_{m \rightarrow s_s} + \tilde{v}_{m \rightarrow s_s}(i) \quad \text{for} \quad m \neq s_s \tag{3.46}$$

Note that $\hat{b}_{m\to s_s}(i)$ can be interpreted as realization of an approximately Gaussian random
variable $\hat{b}_{m\to s_s}$.

4) Average Single-Hop Measurements across Subcarriers

Average the measured single-hop reciprocity coefficients $\hat{b}_{m\to s_s}(i)$ across all sub-carriers to obtain more reliable average (single-hop) estimates $\bar{b}_{m\to s_s}$.

$$\bar{b}_{m\to s_s} = \frac{1}{N_{sc}}\sum_{i=1}^{N_{sc}} \hat{b}_{m\to s_s}(i) \tag{3.47}$$

$$\mathrm{Var}\{\hat{b}_{m\to r_r}\} \approx \sigma^2_{\bar{b}_{m\to r_r}} = \frac{1}{N_{sc}-1}\sum_{i=1}^{N_{sc}} \left|\hat{b}_{m\to r_r}(i) - \bar{b}_{m\to r_r}\right|^2 \tag{3.48}$$

The variance of $\bar{b}_{m\to r_r}$, accounting for the averaging over N_{SC} subcarriers, can be estimated from (3.48) as follows

$$\mathrm{Var}\{\bar{b}_{m\to s_s}\} = \frac{1}{N_{sc}}\ \mathrm{Var}\{\hat{b}_{m\to s_s}\} \approx \sigma^2_{\bar{b}_{m\to s_s}} = \frac{1}{N_{sc}}\sigma^2_{\hat{b}_{m\to s_s}}. \tag{3.49}$$

5) Compute Estimates of Single-Hop Calibration Coefficients Between Reference Antennas

Figure 3-6 shows that the algorithm exploits all single hops between reference antennas in decreasing index order, that is, the hops $s_4 \to s_3$, $s_3 \to s_2$, and $s_2 \to s_1$ are used in the example. Thus, it is important to use reliable estimates for the calibration coefficient for each of these hops between reference antennas.

Consider computing the $\bar{b}_{s_s\to s_l}$ over two hops, using base station antenna m, that is, $s_s \to m \to s_l$, that is,

$$\bar{b}_{s_s\to s_l|m} = \bar{b}_{s_s\to m}\cdot \bar{b}_{m\to s_l} = \frac{\bar{b}_{m\to s_l}}{\bar{b}_{m\to s_s}}, \forall\, m \neq s_s, s_l. \tag{3.50}$$

Reliability can be achieved by a weighted combination of all $\bar{b}_{s_s\to s_l|m}$. Computing the combiner weights requires knowledge of the variance of $\bar{b}_{s_s\to s_l|m}$, which originates from the ratio of two independent, approximately complex Gaussian distributed random variables. This variance can be approximated as follows [8]

$$\mathrm{Var}\{\bar{b}_{s_s\to s_l|m}\} \approx \sigma^2_{\bar{b}_{s_s\to s_l|m}} = \frac{\left|\bar{b}_{m\to s_l}\right|^2}{\left|\bar{b}_{m\to s_r}\right|^2}\left(\frac{\sigma^2_{\bar{b}_{m\to s_l}}}{\left|\bar{b}_{m\to s_l}\right|^2} + \frac{\sigma^2_{\bar{b}_{m\to s_s}}}{\left|\bar{b}_{m\to s_s}\right|^2}\right). \tag{3.51}$$

Reliable estimates $\bar{b}_{s_s\to s_l}$ are obtained by combining

$$\bar{b}_{s_s \to s_l} = \sum_{m=1, m \neq s_s, s_l}^{M} w_m \cdot \bar{b}_{s_s \to s_l | m}, \tag{3.52}$$

with

$$w_m = \frac{1/\sigma^2_{\bar{b}_{s_s \to s_l | m}}}{\sum_{m'=1, m' \neq s_s, s_l}^{M} 1/\sigma^2_{\bar{b}_{s_s \to s_l | m}}}. \tag{3.53}$$

For further use, we derive the variance of $\bar{b}_{s_s \to s_l}$

$$\sigma^2_{\bar{b}_{s_s \to s_l}} = \sum_{m=1, m \neq s_s, s_l}^{M} w_m^2 \cdot \sigma^2_{\bar{b}_{s_s \to s_l | m}}. \tag{3.54}$$

6) Compute Estimates of Calibration Coefficients for all Paths Between BS Antenna m and the Reference Antenna s_1

Based on the results in steps 4 and 5, we compute the calibration coefficients $\bar{b}_{m \to s_1 | \mathcal{A}_n}$ for all sets $\mathcal{A}_n$ as defined in step 2).

$$\bar{b}_{m \to s_1 | \mathcal{A}_n} = \prod_{\forall (a,b) \in \mathcal{A}_s} \bar{b}_{a \to b} = \bar{b}_{m \to s_s} \cdot \prod_{l=2}^{s} \bar{b}_{s_l \to s_{l-1}} \tag{3.55}$$

Note each set $\mathcal{A}_n$ represents one path between antenna m and reference antenna s_1.

With the assumptions that $|\bar{b}_{a \to b}| \approx 1$ and that any potential mutual correlation between the multiplicands in (3.55) can be neglected, the variance of the different mean multi-hop estimates $\bar{b}_{m \to s_1 | \mathcal{A}_n}$ can be approximated by

$$\sigma^2_{\bar{b}_{m \to s_1 | \mathcal{A}_n}} \approx \sum_{\forall (a,b) \in \mathcal{A}_n} \sigma^2_{\bar{b}_{a \to b}} = \sigma^2_{\bar{b}_{m \to s_s}} + \sum_{l=2}^{r} \sigma^2_{\bar{b}_{r_l \to r_{l-1}}}. \tag{3.56}$$

Note that no further processing is needed for the single-hop path between antenna m and reference antenna s_1, which is computed in step 4.

7) Combine Estimates of Calibration Coefficients for all Paths Between BS Antenna m and the Reference Antenna s_1

The purpose of this step is to combine all N estimates $\bar{b}_{m \to s_1 | \mathcal{A}_n}$ into a single reliable estimate $\bar{\bar{b}}_{m \to s_1}$.

$$\bar{\bar{b}}_{m \to s_1} = \sum_{n=1}^{N} w_n \cdot \bar{b}_{m \to s_1 | \mathcal{A}_n} \tag{3.57}$$

The weighting coefficients are coupled to the variance of the individual paths as shown below

$$w_n = \frac{1/\sigma^2_{\bar{b}_{m\to s_1|\mathcal{A}_n}}}{\sum_{n'=1}^{N} 1/\sigma^2_{\bar{b}_{m\to s_1|\mathcal{A}_{n'}}}} \qquad \text{for} \quad m \neq s_1. \tag{3.58}$$

(3.57) represents the final calibration coefficients which are used to calibrate the downlink precoding operation as described in the next subsection.

3.6.5 Reciprocity Correction in the Downlink After Precoding

The reciprocity calibration coefficients, which have been estimated according to the procedure described in the previous section, need to be applied to the uplink channel measurement to compute the precoding matrix. This operation can be written in vector-matrix notation as shown below (single subcarrier).

$$\boldsymbol{H}_{DL}\boldsymbol{B} = \begin{bmatrix} H_{11} & \cdots & H_{1M} \\ \vdots & \ddots & \vdots \\ H_{1K} & \cdots & H_{KM} \end{bmatrix} \begin{bmatrix} \bar{\bar{b}}_{1\to s_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \bar{\bar{b}}_{M\to s_1} \end{bmatrix}. \tag{3.59}$$

Channel coefficients corresponding to the same BS antenna m but different mobile station antenna k are multiplied with the same calibration coefficient. This BS-internal relative reciprocity calibration results in the diagonal matrix $\boldsymbol{B}$.

Tests on the MIMO Prototyping System have shown that the magnitude of $\bar{\bar{b}}_{M\to s_1}$ is close to one. Hence the following approximation holds.

$$\bar{\bar{b}}'_{m\to s_1} = e^{j\text{angle}(\bar{\bar{b}}_{m\to s_1})} \approx \bar{\bar{b}}_{m\to s_1}. \tag{3.60}$$

By defining a matrix $\boldsymbol{B}'$, which contains coefficients $\bar{\bar{b}}'_{m\to s_1}$ on its main diagonal, and inserting it into the precoding matrix definitions (3.18), (3.19), and (3.20), it can be shown that the reciprocity coefficients can be applied to the precoded signal per transmit antenna. Consider the ZF precoding matrix in (3.19) as an example. Inserting $\boldsymbol{B}'$ according to (3.59) yields

$$\widetilde{\boldsymbol{F}}_{ZF} = \beta \boldsymbol{B}'^{H} \boldsymbol{H}_{DL}^{H} (\boldsymbol{H}_{DL}\boldsymbol{B}'\boldsymbol{B}'^{H}\boldsymbol{H}_{DL}^{H})^{-1} = \boldsymbol{B}'^{H}\beta \boldsymbol{H}_{DL}^{H} (\boldsymbol{H}_{DL}\boldsymbol{H}_{DL}^{H})^{-1} = \boldsymbol{B}'^{H}\boldsymbol{F}_{ZF}. \tag{3.61}$$

In (3.61) the fact that $\boldsymbol{B}'\boldsymbol{B}'^{H} = \boldsymbol{I}$ holds because of normalizing the magnitude of all calibrations to 1 as shown in (3.60) has been exploited.

That is, the reciprocity calibration coefficients do not need be considered while computing the precoding matrix. They can be applied after the precoding operation per transmit antenna. The downlink transmission equation (3.1) can hence be extended as follows

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{B}'^{H}\boldsymbol{F}\boldsymbol{x} + \boldsymbol{v}. \tag{3.62}$$

Note that the matrix $\boldsymbol{B}'$ is frequency-independent. That is, the same matrix is applied to the precoded signal at all subcarriers.

3.7 Synchronization and Tracking at the Mobile Station

The application framework ensures that the base station signal processing, as well as the base station radios, are time and frequency synchronous. The BS acts as a time and frequency reference. Mobile stations synchronize their reception to this reference. Likewise, mobile stations adapt time and frequency of their uplink transmission with respect to the base station reference.

Synchronization at the mobile station is comprised of the following two steps in sequence:

1. Initial detection of a receive signal including initial coarse time and frequency synchronization (Time/Freq Sync block in Figure 3-2 and Figure 3-3).
2. Continuous update of initial time and frequency estimates by a tracking algorithm (Freq. Track and Time Track blocks in Figure 3-2 and Figure 3-3).

Both algorithms are presented in this section.

3.7.1 Initial Time and Frequency Synchronization

Initial time synchronization is achieved by a combination of autocorrelation and cross-correlation. Cross-correlation of the received signal with the PSS determines the radio frame start. Cyclic prefix-based auto correlation of the received signal determines the OFDM symbol start. The main principle is depicted in Figure 3-7.

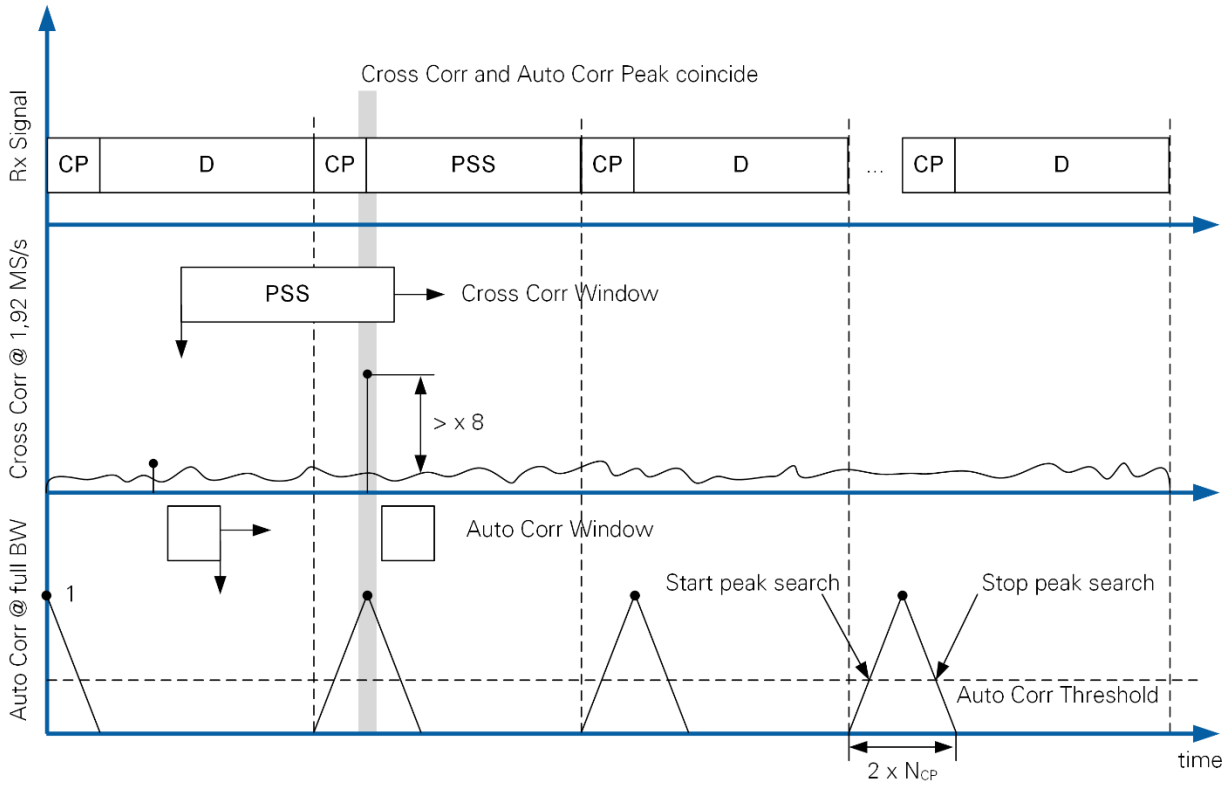

Figure 3-7. Initial Synchronization Principle

PSS-based Cross Correlation

The position of the PSS within a radio frame is known at the mobile station. The cross-correlation of received signal and transmitted PSS sequence can be used to determine the radio frame start as follows.

- The received signal is downsampled by a factor of 16 to compute the cross-correlation at a reduced sampling rate. This operation is valid as the PSS signal uses 62 subcarriers, that is, it has a bandwidth of 1.92 MHz.
- The received signal is constantly cross-correlated with the transmitted PSS. The cross-correlation window length is 2048/16=128.
- The peak value of the cross-correlation is searched within the duration of a radio frame. A new peak search is issued once per radio frame duration.
- The cross-correlation peak is validated to avoid false detections as follows:
    - The cross-correlation result at the potential peak position needs to exceed 8x the average cross-correlation result computed in the corresponding radio frame duration.
    - A valid received radio frame is declared to be detected if valid peaks are found in 3 consecutive radio frame periods. The distance between these peaks may vary no more than ± 5 samples at the reduced sampling rate of 1.92 MS/s, corresponding to 80 samples at 30.72 MS/s

CP-based Autocorrelation

The normalized autocorrelation of the CP with the end of the respective OFDM symbol is constantly computed based on the received signal $y(t)$ according to the metric

$$\Phi(\mathrm{t}) = \left(\frac{\sum_{\tau=1}^{N_{CP}} y(t+\tau) y^*(t - N_{FFT} + \tau)}{\frac{1}{2}\left(\sum_{\tau=1}^{N_{CP}} |y(t+\tau)|^2 + \sum_{\tau=1}^{N_{CP}} |y(t - N_{FFT} + \tau)|^2\right)} \right)^2 \quad (3.63)$$

This metric normalizes the autocorrelation by the energy comprised in the two autocorrelation windows. By design, the magnitude of the metric is equal to one in noiseless frequency flat channels. Note that the correlation window length equals the CP length N_{CP}. The distance of the correlation window equals the FFT size N_{FFT}. The OFDM symbol start is determined as follows:

- The metric is assumed to be computed within the CP if it exceeds a threshold.
- The OFDM symbol start index is searched if
    - The metric exceeds the threshold (*Start peak search* in Figure 3-7). From this point on, the OFDM symbol start index is updated whenever the current metric value exceeds a previous value since the threshold has been surpassed.
    - The OFDM symbol start index search is stopped once the metric falls below the threshold (*Stop peak search* in Figure 3-7). The maximal metric value is reset in this case.
    - The OFDM symbol start index found in the previous two steps is considered valid if the metric exceeded the threshold for at least 8 consecutive samples.

Combining Cross-correlation and Autocorrelation

A valid signal is declared to be received if autocorrelation and cross-correlation both delivered valid results and the OFDM symbol start estimated through cross-correlation and through autocorrelation are equal up to ± 16 samples (that is, ± 1 samples at 1.92 MS/s sampling rate).

The estimated OFDM symbol start corresponds to the first sample of the PSS as shown in Figure 3-8.

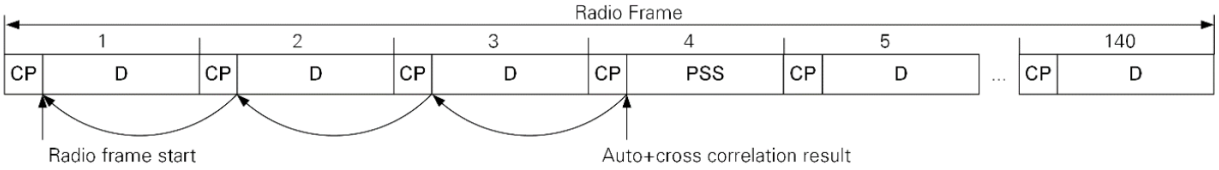

Figure 3-8. Determining the radio frame start from autocorrelation and cross-correlation results.

Assume the PSS is transmitted in the A-th OFDM symbol of a radio frame. The first sample of the first OFDM symbol can be computed as shown below.

$$\mathrm{Idx_{RF}} = \mathrm{Idx_{Corr}} - (\mathrm{A}-1)(N_{FFT} + N_{CP}) - \Delta\,. \tag{3.64}$$

Δ is a configurable margin which shifts the OFDM symbol start into the CP. This margin minimizes the risk of inter OFDM symbol interference in case the estimated OFDM symbol start did not correspond to the first sample of the OFDM symbol.

Carrier Frequency Offset Estimation

Consider the autocorrelation metric in (3.63) at an index t_{max} corresponding to perfect timing synchronization. Also, assume transmission over an additive white Gaussian noise (AWGN) channel in the infinitely high SNR regime. The metric represents the autocorrelation of the transmitted CP with the transmitted last part of the respective OFDM symbol. The enumerator $\Phi_{\mathrm{e}}(\mathrm{t})$ of (3.63) can be rewritten as follows

$$\begin{aligned}\Phi_{\mathrm{e}}(\mathrm{t}_{max}) &= \sum_{\tau=1}^{N_{CP}} x(\tau) e^{j2\pi\Delta f T_s (t_{max}+\tau)} x^*(\tau)\, e^{j2\pi\Delta f T_s (t_{max}+\tau-N_{FFT})} \\ &= e^{j2\pi\Delta f T_s N_{FFT}} \sum_{\tau=1}^{N_{CP}} |x(\tau)|^2 \;= e^{j2\pi\Delta f / f_{sc}} \sum_{\tau=1}^{N_{CP}} |x(\tau)|^2.\end{aligned} \tag{3.65}$$

T_S and f_{SC} denote the sampling duration and the subcarrier spacing, respectively. The carrier frequency offset Δf is estimated from the angle of $\Phi_{\mathrm{e}}(\mathrm{t})$ as follows.

$$\Delta\mathrm{f} = \frac{\mathrm{angle}\left(\Phi_{\mathrm{e}}(\mathrm{t}_{max})\right) f_{SC}}{2\pi}. \tag{3.66}$$

The estimate becomes ambiguous for $\mathrm{angle}\left(\Phi_{\mathrm{e}}(\mathrm{t}_{max})\right) \geq \pi$. Hence, the carrier frequency offset must not exceed half the subcarrier spacing, that is, $|\Delta f| < \frac{f_{SC}}{2}$, such that the estimate is unambiguous.

3.7.2 Time and Frequency Tracking

The initial synchronization results are used as a starting point for further refinement (tracking) once initial synchronization has been successfully achieved. The carrier frequency offset as well as the start of an OFDM symbol are only coarsely estimated during the initial synchronization step. In addition, both parameters may drift over time as a matter of sampling clock offsets between base station and mobile station or temperature changes, for example.

Time and frequency tracking provide an updated time and frequency synchronization estimate once per radio frame. Tracking estimates obtained in the current radio frame are used to constantly adjust time synchronization and frequency correction functions in the beginning of the next radio frame. Both functions rely on evaluating the phase of the channel estimate obtained from pilot OFDM symbols as shown in Figure 3-9.

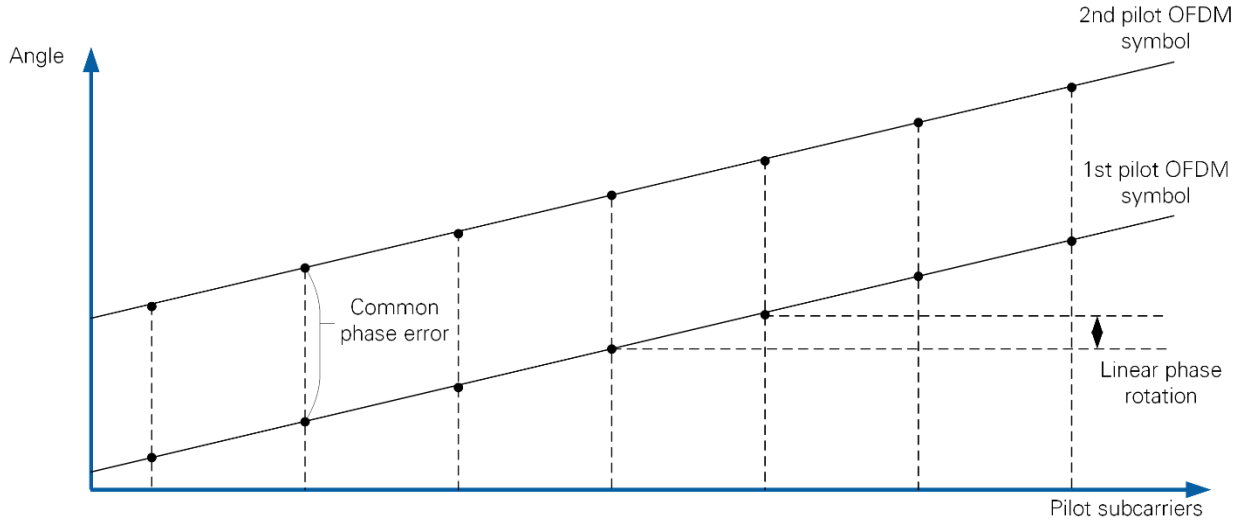

Figure 3-9. Impact of Residual Time and Frequency Errors on the Phase of the Channel Estimate

Frequency Tracking

Frequency tracking is based on estimating the phase difference between two estimated channel transfer functions which have been obtained from two pilot OFDM symbols in the same radio frame (common phase error in Figure 3-9).

Assume the channel to be constant during the first and second pilot OFDM symbol. Also, assume that the channel has been estimated perfectly at pilot subcarrier positions according to (3.6). The common phase error can be estimated by multiplying the channel estimate $\hat{H}_{\mathrm{p},1}(\mathrm{i_p})$ obtained in OFDM symbol 1 and $\hat{H}_{\mathrm{p},2}(\mathrm{i_p})$ obtained in OFDM symbol 2 on subcarrier i_p as follows

$$\hat{\mathrm{H}}_{\mathrm{p},2}(\mathrm{i_p}) \cdot \hat{\mathrm{H}}^{*}{}_{\mathrm{p},1}(\mathrm{i_p}) = \left|H(i_p)\right|^2 \mathrm{e}^{j2\pi\Delta f_{res} T_S \Delta T}. \tag{3.67}$$

ΔT denotes the number of samples between the start of the first and the second pilot OFDM symbol. Computing the sum over all pilot subcarriers

$$\sum_{i_p} \widehat{\mathrm{H}}_{\mathrm{p},2}(\mathrm{i_p}) \cdot \widehat{\mathrm{H}}^{*}{}_{\mathrm{p},1}(\mathrm{i_p}) = \mathrm{e}^{j2\pi\Delta f_{res} T_s \Delta T} \sum_{i_p} \left|H(i_p)\right|^2 \tag{3.68}$$

helps reducing the impact of estimation noise. An estimate of the residual frequency offset can be derived from the angle Φ of (3.68) as shown below

$$\Delta f_{res} = \frac{\Phi}{2\pi T_s \Delta T}. \tag{3.69}$$

The updated carrier frequency offset $\Delta\mathrm{f}(u)$, where u is the radio frame index, can be derived recursively from the carrier frequency offset computed in the previous radio frame u-1 as follows

$$\Delta\mathrm{f}(u) = \Delta\mathrm{f}(u-1) + \lfloor \Delta f_{res}(u) \rceil \tag{3.70}$$

Note that $\Delta\mathrm{f}(u-1)$ has been applied in the correction step before $\Delta f_{res}(u)$ was estimated. $\lfloor \Delta f_{res}(u) \rceil$ indicates a coercion of the estimated residual frequency offset used for correction. This coercion is introduced to limit the correction step size as shown below.

$$\lfloor \Delta f_{res}(u) \rceil = \begin{cases} 10\ Hz, & \Delta f_{res}(u) > 10Hz \\ 0\ Hz, & \text{else} \\ -10\ Hz, & \Delta f_{res}(u) < -10Hz \end{cases}. \tag{3.71}$$

The received time domain signal $y'(t)$ is frequency-shifted as shown below, starting from the first sample of the following radio frame $u+1$

$$y(t) = y'(t) \cdot\ e^{-j\frac{2\pi\Delta f(u)}{f_s}t}. \tag{3.72}$$

Note that the estimate $\Delta f_{res}(u+1)$ is computed based on the corrected signal $y(t)$ (3.72).

Time Tracking

Imperfect time synchronization results in a residual time shift a_{res}, representing a fraction of the sampling interval, of the FFT window which causes a linear phase rotation over frequency: $x_{time}(t - a_{res}T_s) \leftrightarrow x_{freq}(i)e^{-j\frac{2\pi i a_{res}}{N_{FFT}}}$ (see Figure 3-9). Shifting the FFT window into the CP ($a_{res} > 0$) causes a linearly decreasing phase with increasing subcarrier index, for instance.

The time shift a_{res} can be derived from the phase difference between the channel measured at two adjacent pilot subcarriers $i_{p,2} > i_{p,1}$

$$\Omega(\mathrm{i}_{p,1}, \mathrm{i}_{p,2}) = \mathrm{angle}\left(\widehat{\mathrm{H}}_{\mathrm{p}}(\mathrm{i}_{\mathrm{p},2}) \cdot \widehat{\mathrm{H}}^{*}{}_{\mathrm{p}}(\mathrm{i}_{\mathrm{p},1})\right) = \frac{2\pi a_{res}(i_{p,1} - i_{p,2})}{N_{FFT}}. \tag{3.73}$$

A pilot OFDM symbol comprises $N_{Pil} = 100$ pilot symbols with a spacing of 12 subcarriers between pilot symbols. The aggregate phase rotation over all pilot subcarriers

can be derived from the phase rotation between two adjacent pilot subcarriers as shown below.

$$\Omega\left(\mathrm{i}_{p,1},\mathrm{i}_{p,N_{Pil}}\right)=\sum_{l=1}^{N_{Pil}-1}\Omega\left(\mathrm{i}_{p,l},\mathrm{i}_{p,l+1}\right)=\frac{2\pi a(i_{p,1}-i_{p,N_{Pil}})}{N_{FFT}}=\frac{2\pi\cdot(i_{p,1}-i_{p,N_{Pil}})}{N_{FFT}}a_{res}. \tag{3.74}$$

Finally, the residual time shift follows from the aggregate phase rotation as

$$a_{res}=\frac{\Omega\left(\mathrm{i}_{p,1},\mathrm{i}_{p,N_{Pil}}\right)\cdot N_{FFT}}{2\pi\cdot(i_{p,1}-i_{p,N_{Pil}})}. \tag{3.75}$$

Note that the spacing between first and outmost pilot subcarrier is $\left(i_{p,1}-i_{p,N_{Pil}}\right)=1188$ for the frequency orthogonal pilot design used in the application framework.

The updated time shift $a(u)$ can be derived recursively from the time shift computed in the previous radio frame u-1 as follows

$$a(u)=a(u-1)+\lfloor a_{\mathrm{res}}(u)\rfloor. \tag{3.76}$$

Note that $a(u-1)$ has been applied in the correction step before $a_{res}(u)$ was estimated. $\lfloor a_{res}(u)\rfloor$ indicates a coercion of the estimated time shift used for correction. This coercion is introduced to limit the correction step size as shown below

$$\lfloor a_{res}(u)\rfloor=\begin{cases}1, & a_{res}(u)>1\\ 0, & \text{else}\\ -1, & a_{res}(u)<-1\end{cases}. \tag{3.77}$$

The received time domain signal $y'(t)$ is time-shifted as shown below, starting from the first sample of the following radio frame $u+1$

$$y(t)=y'(t+a(u)). \tag{3.78}$$

Note that the estimate $a_{res}(u+1)$ is computed based on the corrected signal $y(t)$ (3.78).

3.8 Receive AGC and Open Loop Uplink Power Control

3.8.1 Receive AGC

The receive gain control adjusts analog gain stages to provide a proper (target) signal level at the input to the analog-to-digital convertor (ADC). The target input level to the ADC needs to be chosen sufficiently high to alleviate the impact of quantization noise. On the other hand, it needs to be sufficiently low to prevent clipping of the receive signal at the ADC. Note that the target input level to the ADC translates into a signal power at the ADC output which can be measured in digital baseband. Thus, the analog gain control (AGC) adjusts the receive gain to achieve a target power at the ADC output.

The AGC algorithm is typically implemented at the mobile station downlink receiver. Uplink power control guarantees a proper signal level at the base station uplink receiver, such that AGC functionality does not necessarily need to be implemented at the base

station. The accuracy of the open loop uplink power control functionality provided with the application framework is limited. Therefore, AGC functionality is implemented also at the base station side.

Note that multi-antenna stations (BS and multi-antenna MS) can use the gain range {0,…31.5} dB. The single-antenna MS can use the gain range {0,…,37.5} dB. The difference is due to implementation constraints.

Two different operation modes are distinguished, depending on the connection state. In each of these operation modes, a single common receive gain value is chosen for all antennas.

Unsynchronized State

Prior to establishing a connection, no information is available about the receive signal level. The receive signal may originate from thermal receiver noise or from a transmitted signal. The AGC algorithm periodically sweeps through the available gain range as shown in Figure 3-10, in this case.

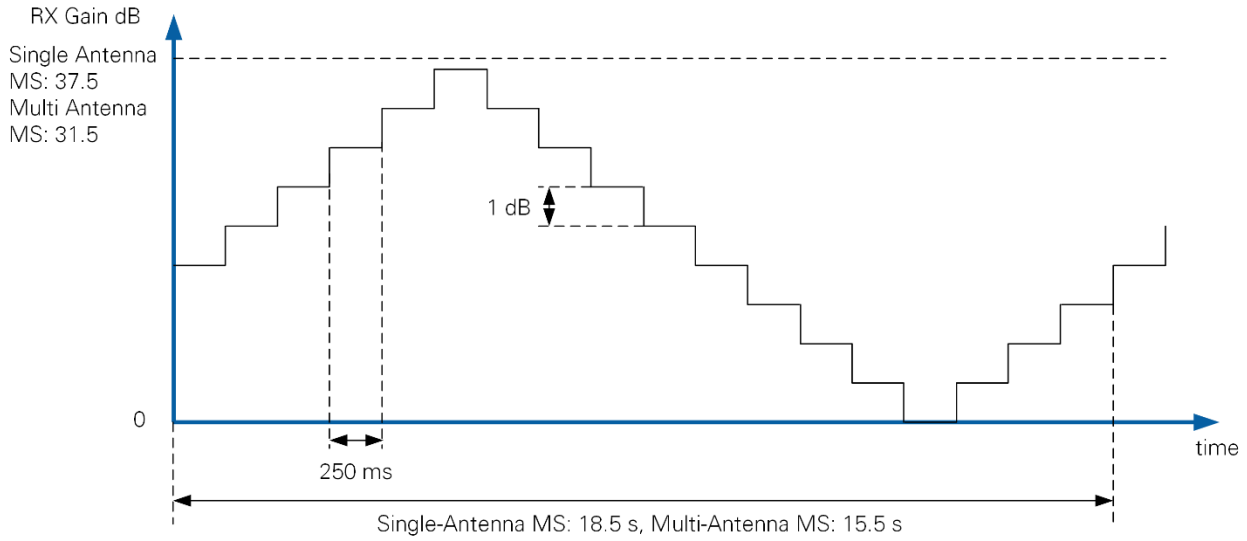

Figure 3-10. Receive AGC Principle before Synchronization

Testing the complete gain range ensures that signal sources which are placed very close to or very far from the receiver can be detected by the synchronization algorithm.

Note that only mobile stations start in unsynchronized state. The base station starts in synchronized state already.

Synchronized State

In synchronized state, the synchronization algorithm at the mobile station has synchronized in time and frequency to the base station. Likewise, at the base station a valid uplink receive signal (presence of a spatial layer) has been detected as described in section 3.9.2.

In synchronized state, the receiver measures the receive power in digital baseband in each OFDM symbol of a radio frame at all receive antennas. The maximum symbol power across OFDM symbols and antennas is used to adjust the receive gain. Note that the maximum receive power will typically be received in a precoded data OFDM symbol at the mobile station in the downlink. The receive gain is adapted as follows:

- If the receive power is smaller than the target power: increment the receive gain by one gain step.
- If the receive power exceeds the target power: decrement the receive gain by one gain step.

A gain step of 0.5 dB has been pre-configured at base station and mobile stations.

3.8.2 Open Loop Uplink Transmit Power Control

Uplink transmit power control is implemented at the mobile station to ensure that the uplink signals are received at the base station at a target power level. The application framework does not implement control channels. Hence, no transmit power adjustment information can be sent from the base station to the mobile stations.
Under these conditions, an open loop uplink power control algorithm is implemented. It is based on the following principles

- Knowledge of base station receive gain and transmit power settings is made available at the mobile station by the user. The number of active mobile stations as well as the number of base station antennas is known at each mobile station as well.
- The receive power, measured for the (un-precoded) PSS at the mobile station is used in combination with a link budget analysis to derive the uplink transmit power.
- A single common transmit power value is adjusted for all transmit antennas of the mobile station.

Link Budget Model

Open loop power control relies on the link budget. Figure 3-11 shows the link budget model for uplink and downlink.

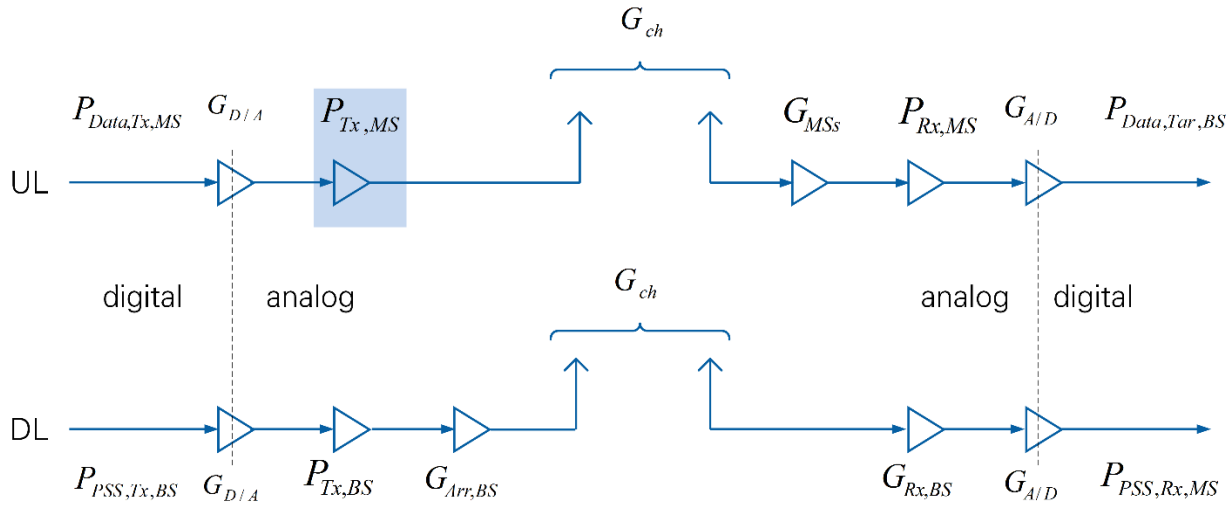

Figure 3-11. Link Budget Model

The different components encompassed in this model are as follows (from the left to the right):

- $P_{Data,Tx,MS}$: transmit power of a data OFDM symbol, $P_{Data,Tx,MS} = -17\ dBFs$
- $P_{PSS,Tx,BS}$: transmit power of a PSS OFDM symbol, $P_{PSS,Tx,BS} = -30dBFs$
- $G_{D/A}$: gain of the digital-to-analog convertor (DAC)
- $P_{Tx,MS}$: transmit reference power level set at the mobile station
- $P_{Tx,BS}$: transmit reference power level set at the base station
- $G_{Arr,BS}$: base station array gain. The PSS is transmitted from all base station antennas in an un-precoded fashion. The array gain depends on the (unknown) position of the mobile station relative to the base station array, as well as the array geometry. The average array gain is approximated by $G_{Arr,BS} = 10\log(M)$, which corresponds to the gain in receive power by superimposing M uncorrelated signals. Note that the maximum possible gain $G_{Arr,BS} = 20\log(M)$ would result for a mobile station antenna placed at a position where all M signals emitted from the base station superimpose constructively. Note that the BS array gain is a large unknown. For instance, for 16 BS antennas it could vary from less than 12 dB up to 24 dB.
- G_{ch}: gain of the wireless channel
- G_{MSs}: The signal received at each base station antenna originates from K mobile station antennas, transmitting uncorrelated signals. Assuming all mobile stations have adjusted their transmit power, the receive power at the base station is $G_{MSs} = 10\log(K)$ larger as compared to receiving a signal from a single mobile station antenna.
- $P_{Rx,MS}$: receive reference power level at the mobile station
- $G_{RX,BS}$: receive gain at the base station
- $G_{A/D}$: gain of the ADC
- $P_{Data,Tar,BS}$: target receive power of data OFDM symbols at a base station antenna

- $P_{PSS,Rx,MS}$: receive power of the PSS OFDM symbol at a station antenna

Relation of Reference Power Levels and Gain

- Receive reference power level: the receive gain is computed such that a sinusoidal signal received at this power would optimally excite the ADC. A decrease in receive reference power level by X dB leads to an increase in receive gain by X dB.
- Transmit reference power level: the transmit gain is adjusted such that a sinusoidal signal which fully excites the DAC would result in this transmit power. An increase in transmit reference power level by X dB leads to an increase in transmit gain by X dB

Uplink Transmit Power Computation

We start by formulating the receive power at mobile station and base station using the link budget model in Figure 3-11.

$$P_{Data,Rx,BS} = P_{Data,Tx,MS} + G_{D/A} + P_{Tx,MS} + G_{ch} + G_{MSs} - P_{Rx,MS} + G_{A/D} \overset{!}{=} P_{Data,Tar,BS} \tag{3.79}$$

$$P_{PSS,Rx,MS} = P_{PSS,Tx,BS} + G_{D/A} + P_{Tx,BS} + G_{Arr,BS} + G_{ch} + G_{Rx,BS} + G_{A/D} \tag{3.80}$$

Solving (3.79) and (3.80) for G_{ch}, combining both equations and solving for $P_{Tx,MS}$ yields the transmit power to be adjusted at the mobile station

$$P_{Tx,MS} = P_{Data,Tar,BS} - P_{Data,Tx,MS} - P_{PSS,Rx,MS} + P_{PSS,Tx,BS} + P_{Tx,BS} + G_{Arr,BS} - P_{Rx,MS} - G_{Rx,BS} - G_{MSs}. \tag{3.81}$$

The AGC algorithm is executed before the uplink power control algorithm at the mobile station. Hence, the reference receive power level $P_{Rx,MS}$ is known. The known base station parameters $P_{Tx,BS}$, $G_{Arr,BS}$, $G_{Rx,BS}$, and the number of active mobile station antennas, captured in G_{MSs} must be provided by the user.

The power $P_{PSS,Rx,MS}$ at which the PSS has been received at all antennas of the mobile station is a known input to the uplink power control algorithm. The power control algorithm can be configured to use either of the following two options:

- The maximum power $P_{PSS,Rx,MS}$ measured across all mobile station
- The average power $P_{PSS,Rx,MS}$ measured across all mobile station antennas

The former option will typically result in a smaller transmit power as compared to the latter option.

3.9 Spatial Layer Mapping and Detection

The application framework does not implement control channels to signal the use of certain spatial layers. Enabling the data transmission using one or multiple spatial layers is implemented on the mobile station side. Once a layer is enabled, pilots and data are transmitted on the uplink. The base station detects the presence of a spatial layer in the

uplink, adjusts its uplink and downlink signal processing accordingly, and transmits this layer also on the downlink. Hence, a method is needed to detect layers, which have been switched on and off on the uplink, reliably at the base station.

This section first presents the mapping of layers to antennas in subsection 3.9.1, followed by an introduction of the method to detect the number of active spatial layers at the BS.

3.9.1 Spatial Layer Mapping

The application framework supports up to 12 spatial layers. These could be allocated to 12 single-antenna mobile stations or to a single multi-antenna mobile station. A spatial layer is coupled to a certain pre-configurable modulation scheme in the application framework. Also, a spatial layer is coupled to layer-specific pilot signals in the uplink and in the downlink.

Layer-Specific Pilot Signals

Pilots coupled to different spatial layers are transmitted frequency-orthogonal as presented in section 2.2.1. Pilots coupled to the first spatial layer are transmitted using the first subcarrier in each RB. Pilots coupled to the second spatial layer use the second subcarrier in each RB, and so on. This principle holds for uplink and downlink pilots.

Layer to Antenna Mapping at the Mobile Station

The number of active transmit antennas, that is, antennas used for signal transmission equals the number of spatial layers which have been selected for transmission from this mobile station. Assume the example of an MS with four antennas. Assume the spatial layers (1,3,8) have been selected for transmission. In that case, the first three transmit antennas would be used with the mapping: Layer=1 → Ant 1; Layer=3 → Ant=2; Layer=8 →Ant=3. Assume also the fifth layer would be enabled later in addition, that is, the layers (1,3,5,8) are transmitted. The layer-antenna mapping would change to: Layer=1 → Ant 1; Layer=3 → Ant=2; Layer=5 → Ant = 3; Layer=8 →Ant=4. That is, the fourth layer is now transmitted from antenna 4 instead of antenna 3.

Note that all receive antennas at the mobile station are used for signal reception, regardless of the number of active spatial layers.

Layer to Antenna Mapping at the Base Station:

Upon detecting the use of certain spatial layers, as described in the next sub-section, the base station starts using these exact same layers also for downlink transmission to this mobile station. That is, spatial layer-specific pilots and data are transmitted.

3.9.2 Spatial Layer Detection at the Base Station

A robust method to detect the presence of a spatial layer has been devised in *Algorithm for Detecting the Number of Transmit Antennas in MIMO-OFDM Systems* [9]. It exploits the fact that that pilots coupled to different layers are frequency orthogonal. It further exploits the assumption that the channel impulse response length is short compared to the OFDM symbol length. Two cases, shown in Figure 3-12, can be distinguished:

1. The channel impulse response has been estimated for a layer that was present. The channel impulse response will be in a narrow window, denoted *Channel Window* of length N_{CW}. The remaining part of the estimate of length N_{NW}, denoted *Noise Window*, contains estimation noise.
2. The channel impulse response has been estimated for a layer that was not present. The channel impulse response comprises noise only in both windows.

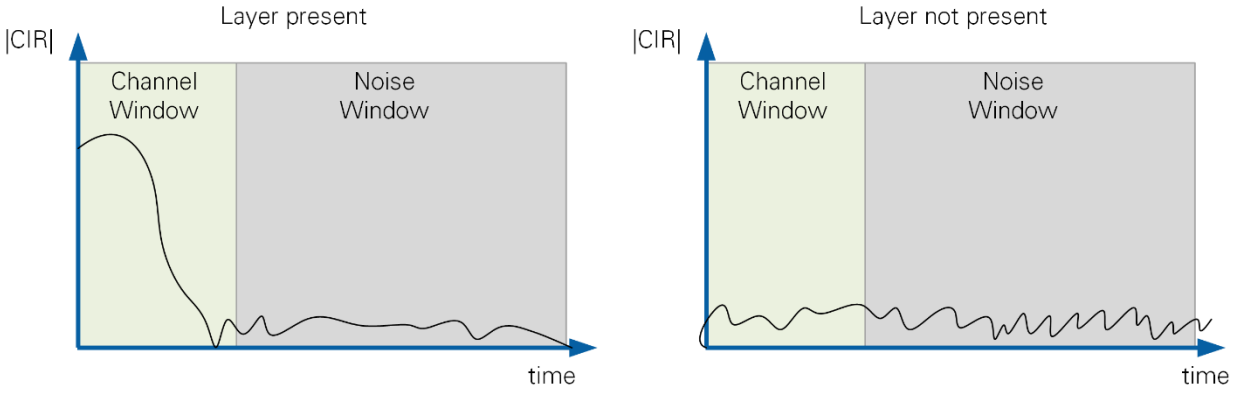

Figure 3-12. Channel Impulse Response Estimated for a Present Layer and a Non-Present Layer

A metric Θ is computed to accept one of the following two hypotheses:

$$\Theta = \frac{\sum_{\tau=1}^{N_{CW}} |CIR(\tau)|^2}{\sum_{\tau=1}^{N_{CW}+N_{NW}} |CIR(\tau)|^2} \approx \frac{SNR + \frac{N_{CW}}{N_{CW}+N_{NW}}}{SNR+1} \quad . \tag{3.82}$$

The right-hand-side of (3.82) relates the metric to the SNR and the window length. This metric converges to 1 at infinitely high SNR and to $\frac{N_{CW}}{N_{CW}+N_{NW}}$ at infinitely low SNR.

The pilot design used in the application framework is localized within 1,200 out of 2,048 subcarriers. One hundred pilot symbols are uniformly distributed with a spacing of 12 subcarriers within the 1,200 subcarriers. Computing the channel transfer function at the 100 used subcarriers and computing the length-100 IDFT from this transfer function yields a length-100 estimate for the channel impulse response. Note that using a bandwidth of 1,200 subcarriers out of 2,048 subcarriers scales the CP length by a factor of $1{,}200/2{,}048$ which yields 84 samples instead 144 samples. That is, the length 100 estimated channel impulse response covers the CP length and 16 extra samples.

The layer detection metric versus SNR is shown in Figure 3-13 for different length of the channel window. Assuming the channel to be shorter than 25% of the CP duration, we can select a threshold of 0.4 to decide about the hypothesis reliably even below and SNR of 0 dB. A layer is declared to be detected if $\Theta > 0.4$. Otherwise, this layer is declared not to be present in the uplink signal.

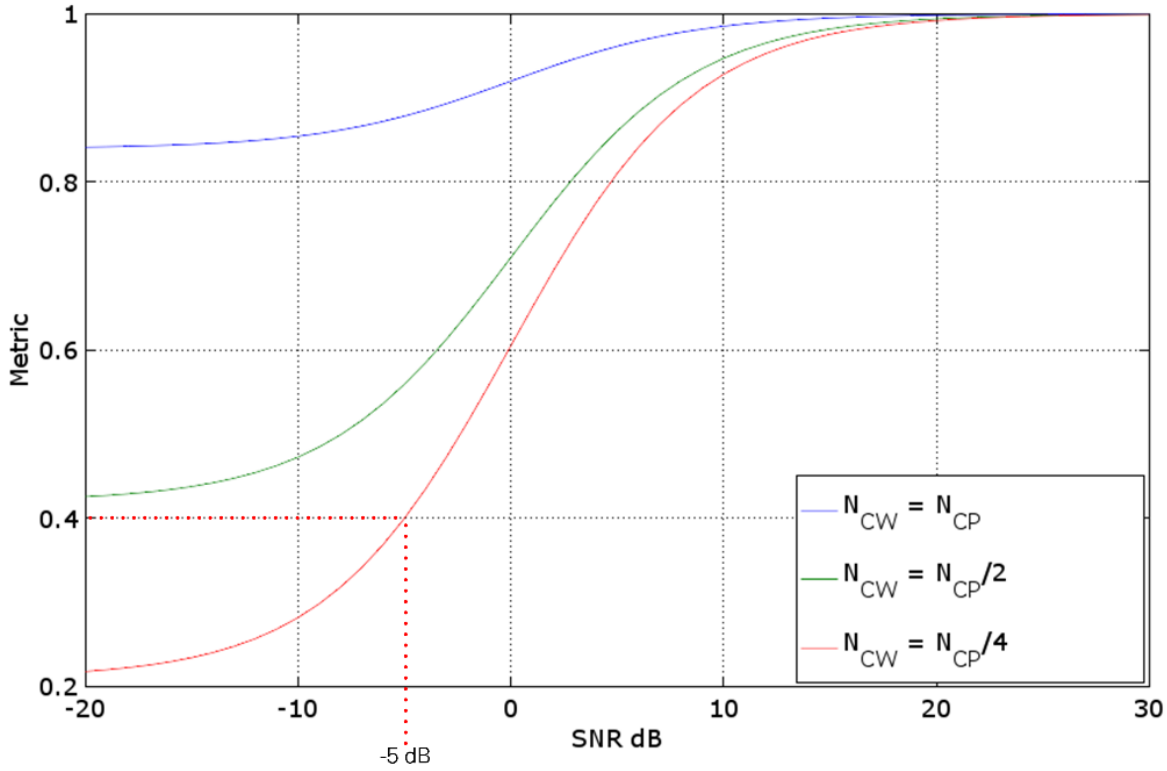

Figure 3-13. Layer Detection Metric

In addition, the received signal power per layer is evaluated as follows:

- For a layer to be declared to be detected, the receive power coupled to this layer must be above -55 dBFs.
- For a layer to be detected, its receive power must not be more than 20 dB below the layer which is received at the highest receive power.

Both the layer detection algorithm and the receive power criterion need to be fulfilled simultaneously for a layer to be detected. If these criteria are not fulfilled but the CRC check for this layer was passed successfully, this layer is still considered successfully detected.

3.10 Scrambling

Transmitted transport blocks are scrambled with a pseudo-random sequence on a per OFDM symbol basis. Likewise, hard-detected bits after demodulation and log-likelihood ratio (LLR) computation are descrambled.

The scrambling sequences are defined by a length-31 Gold sequence. The output sequence $c(n)$ of length M_{PN}, where $0 \leq n \leq M_{PN} - 1$, is defined by

$$\begin{aligned} \mathrm{c}(n) &= \left(x_1(n+N_c)\right) + \left(x_2(n+N_c)\right) \mathrm{mod}\ 2 \\ \mathrm{x}_1(n+31) &= (\mathrm{x}_1(\mathrm{n}+3) + \mathrm{x}_1) \mathrm{mod}\ 2 \\ \mathrm{x}_2(n+31) &= \left(x_{2(n+3)} + x_2(n+2) + x_{2(n+1)} + x_2(n)\right) \mathrm{mod}\ 2 \end{aligned} \quad (3.83)$$

where $N_c = 1{,}600$.

The first m-sequence is initialized with $x_1(0) = 1, x_1(n) = 0, 1 \leq n \leq 30$. The initialization of the second m-sequence is denoted by $c_{init} = \sum_{i=0}^{30} x_2(i) \cdot 2^i$.

The scrambling sequence generator is initialized at the start of each OFDM symbol, where the initialization value c_{init} is given by $c_{init} = n_1 \cdot 2^{23} + n_2 \cdot 2^7 + n_3 \cdot 2^3 + n_4$, where

- n_1 corresponds to the 8-bit representation of the OFDM symbol number [0…139]
- n_2 corresponds to the 16-bit representation of 0
- n_3 corresponds to the 4-bit representation of the spatial layer index [0…11]
- n_4 corresponds to the 3-bit representation of the modulation type [1: QPSK, 2: 16-QAM, 3: 64-QAM, 4: 256-QAM].

3.11 Computation of Channel Quality Metrics

3.11.1 MIMO Channel Singular Values

The MIMO channel singular values can be used as a metric to determine how many spatial layers to use, or to compare different propagation conditions in an experiment.

The application framework uses the estimated frequency domain channel transfer function to compute the MIMO channel singular values as follows.

In a first step, the aggregate frequency domain MIMO channel $\boldsymbol{H}_{aggr}$ is computed as shown below.

$$\boldsymbol{H}_{aggr} = [\boldsymbol{H}_1, \boldsymbol{H}_2, \ldots, \boldsymbol{H}_{N_{PRB}}]. \quad (3.84)$$

$\boldsymbol{H}_{aggr}$ is computed by concatenating the MIMO channel matrices corresponding to different RBs $\boldsymbol{H}_i$ with $1 \leq i \leq N_{PRB}$. The dimension of $\boldsymbol{H}_{aggr}$ at the BS will be 128 x 1,200 in case a 128-antenna base station is used. Note that $\boldsymbol{H}_i$ contains channel estimates for all 12 possible mobile station transmit antennas even though they may not be active.

The singular value decomposition is computed based on the aggregate channel matrix as shown below.

$$\mathbf{USV}^{\mathrm{H}} = \mathrm{svd}\left(\boldsymbol{H}_{aggr}\right). \quad (3.85)$$

The matrix $\boldsymbol{S}$ contains 12 singular values on its main diagonal. These singular values can be interpreted as average singular values which characterize the multi- path MIMO channel [10].

Note that the singular values are computed based on the actual wireless channel at the base station. However, they are computed based on the effective channel $\widetilde{\boldsymbol{H}}$ including precoding at the mobile station. That is, the received downlink pilots which have been precoded at the base station are used at the mobile station to compute the channel transfer function and the singular values.

3.11.2 Error Vector Magnitude

The error vector magnitude or, likewise, the SNR are used as a metric to evaluate the signal quality at the equalizer output.

The transmitted symbol is detected in a first step by computing the Euclidean distance between equalized symbol and all symbols contained in the transmit signal alphabet $\mathcal{X}$

$$x_d = \arg\min_{x' \in \mathcal{X}} |\hat{x} - x'|^2 \,. \tag{3.86}$$

The squared Euclidean distance $|\hat{x} - x_d|^2$ is further used to compute the EVM as follows

$$\mathrm{EVM} = \sqrt{\frac{\frac{1}{N}\sum_{i=1}^{N} |\hat{x}_i - x_{d,i}|^2}{\frac{1}{|\mathcal{X}|}\sum_{x \in \mathcal{X}} |x|^2}}. \tag{3.87}$$

$|\mathcal{X}|$ denotes the cardinality of set $\mathcal{X}$, that is, the number of modulation symbols.

4 Hardware Partitioning and Data Routing

This chapter introduces how the signal processing functionality has been mapped to hardware modules, as well as the hardware setup of the base station and the mobile stations. Important interfaces between hardware modules are also explained.

4.1 Hardware Partitioning and Algorithm Mapping

In section 3.1, we introduced the signal processing blocks of the base station and the mobile stations. This section provides an overview about how this signal processing functionality is mapped to hardware modules. Following sections will introduce the rationale for this mapping, such as interconnection bandwidth and FPGA real estate.

Refer to the *MIMO Prototyping System Getting Started Guide* [2] for a detailed hardware component list an overview over the hardware partitioning and an introduction of the application framework software architecture which is closely aligned to the detailed hardware partitioning presented in this chapter.

4.1.1 Base Station

Section 3.1.1 introduced the base station signal processing. It was already indicated that the signal processing is split into the following three distinct blocks: bit processor (outer transceiver), MIMO processor and remote radio head (RRH) (inner transceiver). This split reflects in the mapping of signal processing functionality to hardware modules, shown in Figure 4-1.

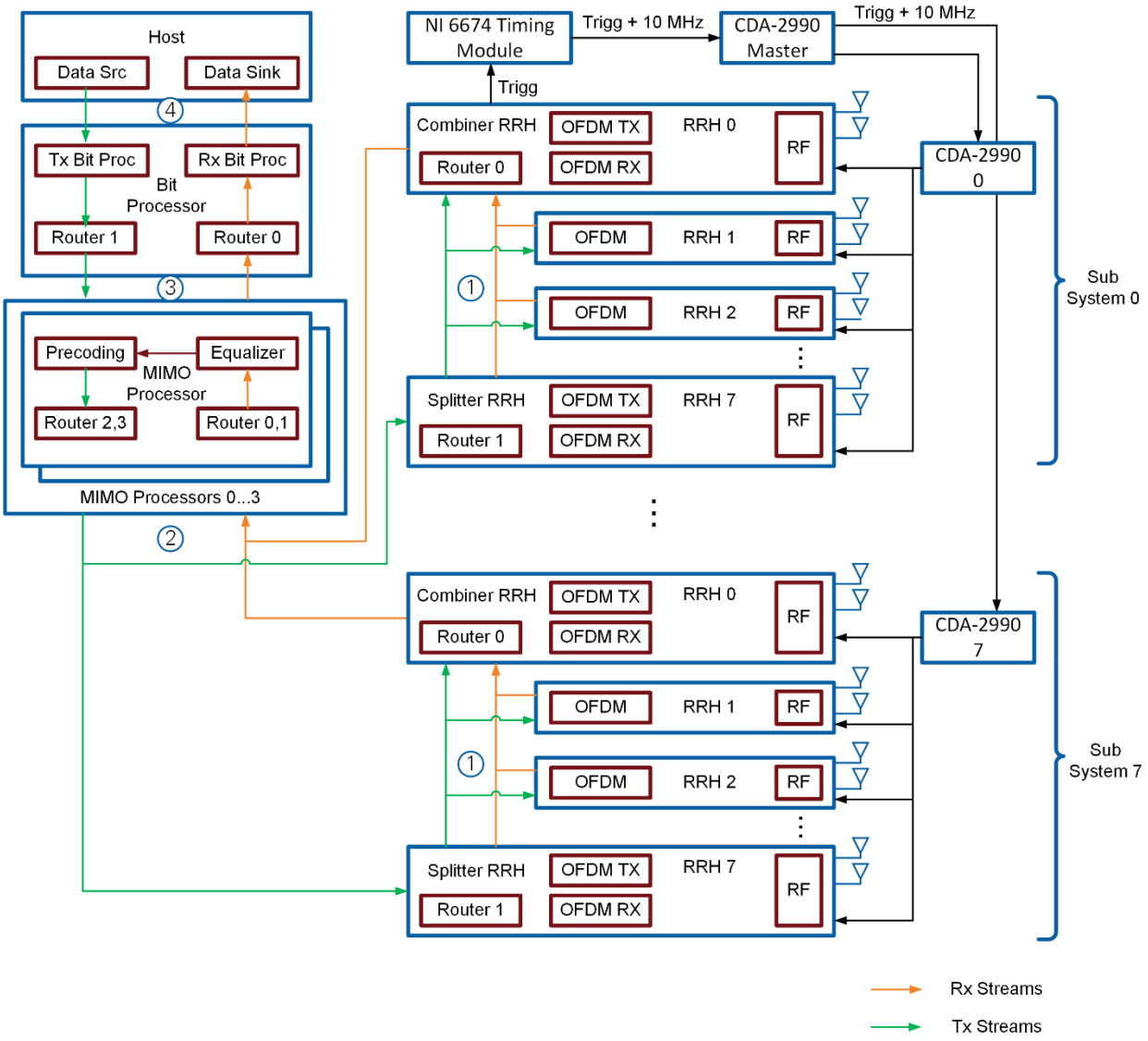

Figure 4-1. Base Station Hardware Architecture (Maximal Configuration, 128 Antennas)

RRH:

RRHs are implemented on USRP devices. Each RRH comprises two antennas, two radio chains, and the respective ADC/DAC modules. It implements OFDM functionality such as (I)FFT or the insertion/removal of cyclic prefixes as well as digital front end functionality such as re-sampling on an FPGA. Eight RRHs form a subsystem which is interconnected through peer-to-peer (P2P) FIFOs (interface 1). Data arriving from the MIMO processors at RRH 7 (splitter RRH) is distributed to other RRHs in the same subsystem over interface 1. Likewise, the output of all RRHs in a subsystem is collected at RHH 0 (combiner subsystem) over interface 1 and forwarded to the MIMO processors. Data collection and distribution is implemented through router modules on RRH 0 and RRH 7. Thus, subsystems can minimize the number of DMA FIFOs at the MIMO processor, which is important as this number is limited.

The number of subsystems depends on the number of antennas. The maximum

configuration comprises eight subsystems, supporting 128 antennas. The minimum configuration comprises one subsystem and 2 antennas. Section 4.2.1 discusses the subsystem design choices.

MIMO Processors:
MIMO Processors are implemented on PXIe-7976 FPGA Module for FlexRIO modules. They comprise MIMO processing operations such as precoding and equalization, as well as channel estimation in frequency domain. The system comprises one to four MIMO processors, depending on the number of antennas, as will be explained in section 4.2.2. Each MIMO processor processes a subset of all subcarriers. The MIMO processor exchanges modulated or equalized QAM symbols with the bit processor through P2P FIFOs (interface 3). It exchanges received frequency domain data or precoded frequency domain transmit data with the RRHs over interface 2.

Bit Processor:
The bit processor is implemented on a PXIe-7976 module. It comprises functionality such as (de)modulation, CRC check and bit packaging. The bit processor exchanges binary data with the host controller over DMA FIFOs (interface 4). It collects equalized data from or distributes modulated data to the MIMO processors over interface 3.

Host:
The host is implemented on a PXIe-8135 FPGA Module for FlexRIO. It comprises data source and sink, as well as configuration and monitoring logic.

Timing and Clock Distribution:
At the base station side, it is important to ensure time and frequency synchronicity of the different RRHs. This is achieved as follows.

- The first RRH (RRH 0) in the first subsystem (subsystem 0) generates a trigger pulse (Trigg) as soon as it is ready to transmit data in the downlink.
- This trigger pulse is fed to the PXIe-6674 Synchronization Module. The timing module amplifies this pulse and provides a high-precision 10 MHz reference signal.
- The pulse and 10 MHz reference signal are forwarded to a master CDA-2990 Clock Distribution Device. This module distributes both signals to up to eight RRH subsystems. Each subsystem is assigned a dedicated CDA-2990 device, which distributes the trigger pulse and the 10 MHz reference to each RRH (including RRH 0 in subsystem 0). The 10 MHz reference is used to derive local oscillator signals for up and down conversion, to derive the ADC/DAC sampling clocks and to derive the FPGA clocks.
  The trigger pulse is used to start FPGA signal processing (transmission) synchronously at all FPGAs. Note that the trigger pulse may still be asynchronously sampled at the different USRP devices. A standard USRP driver functionality ensures that this asynchronous sampling of trigger pulses is compensated for.

4.1.2 Multi-Antenna Mobile Station

The signal processing block diagram of the multi-antenna mobile station has been introduced in Figure 3-3. It shares most signal processing functionality, except for synchronization and precoding with the base station. Figure 4-2 shows the mapping of signal processing functionality to hardware modules.

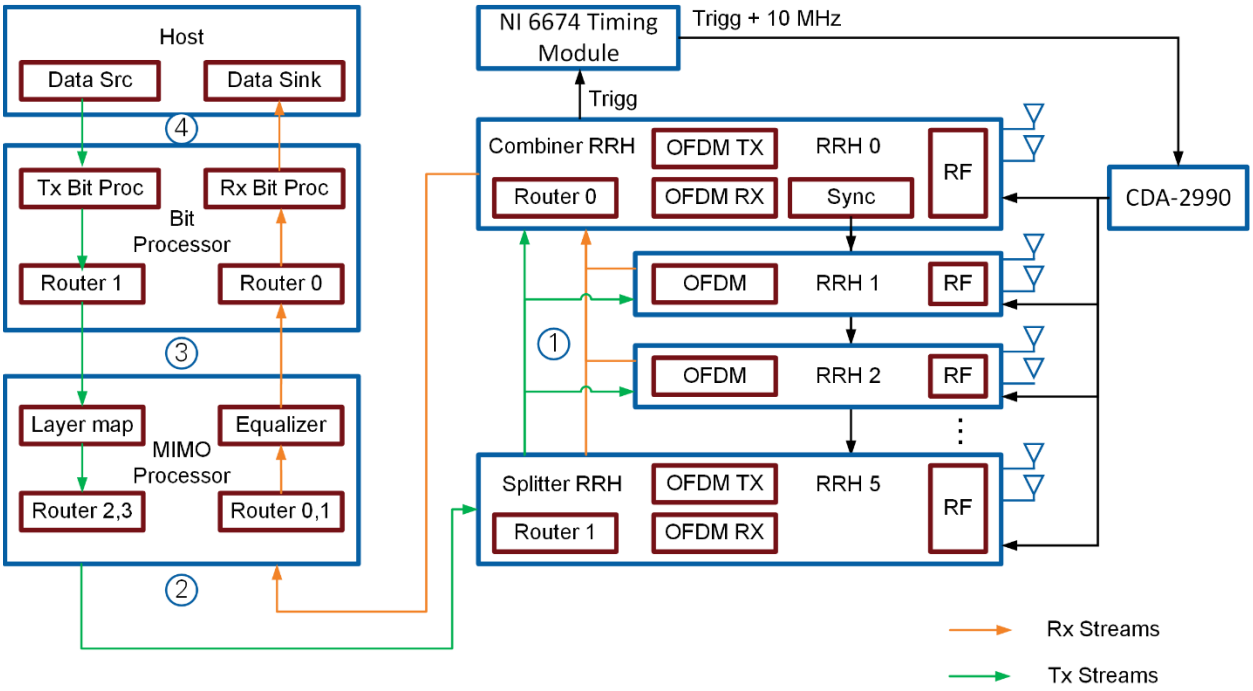

Figure 4-2. Multi-antenna Mobile Station Hardware Architecture (Maximal Configuration, 1 Antenna)

The mapping is identical as compared to the base station. The main differences are as follows:

RRHs: A subsystem comprises of up to 6 RRHs, that is, up to 12 antennas. The synchronization results are shared with all other RRHs in the subsystem and applied synchronously.

MIMO processor: A single MIMO processor processes all subcarriers. Uplink data is not precoded but directly mapped to antennas.

Timing and Clock Distribution: There is a single CDA-2990 device that distributes the trigger pulse and the 10 MHz reference to all RRHs. The trigger pulse ensures that the reception of signal starts synchronously at all RRHs. Also, it starts a counter in the transmit path of each RRH synchronously.

4.1.3 Single Antenna Mobile Station

Figure 3-2 shows the signal processing block diagram of the single antenna mobile station. All functionality is mapped to a single RRH as shown in Figure 4-3.

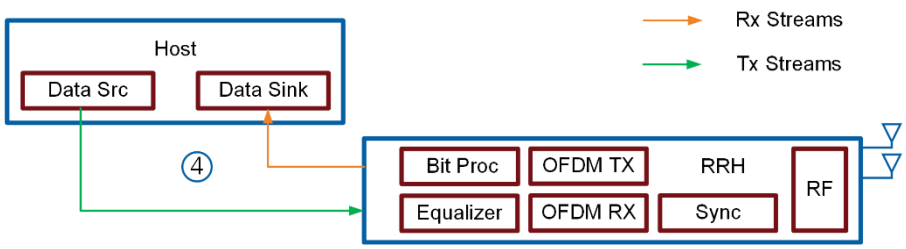

Figure 4-3. Single-antenna Mobile Station Hardware Architecture

That is, the functionality of the MIMO processor (equalization) and the bit processer has been added to the RRH. Note that a single RRH supports two antennas. It also supports the signal processing for two single antenna mobile stations which operate independently but share the same RRH hardware. The RRH exchanges binary data with the host over interface 4 like the base station and the multi-antenna mobile station.

4.2 Platform Considerations

This section provides the background why the hardware partitioning presented in section 4.1 has been chosen. More specifically, we discuss the choice of subsystems, FIFO numbers, backplane rates, the FPGA size and provide rationale for the bit processor FPGA.

4.2.1 Subsystems

Figure 4-1 shows the base station system configuration comprising multiple subsystems of multiple RRHs each. For the maximum number of 128 antennas, 64 RRHs are required in the system. The PCI express connections enable exchanging data between the different devices of one BS or MIMO MS using P2P streams. Each P2P stream requires a DMA FIFO on transmitter and receiver side. The stream enables unidirectional traffic. For serving the transmit and the receive path, two P2P streams are required per RRH.

For MIMO processing, data corresponding to all antennas must be combined on the MIMO processors. Each MIMO processor offers up to 32 DMA FIFOs, which is a restriction of the FPGA module. It is therefore not possible to connect each MIMO processor and each of the up to 64 RRHs directly. To save DMA FIFOs on the MIMO processor for communication to the Host and the bit processor a maximum of 16 DMA FIFOs are used for connection to RRHs (next smaller power of 2).

To connect 64 RRHs to the MIMO processors the hierarchy level of subsystems is introduced. Each subsystem (shown in Figure 4-4) comprises an RRH Combiner which aggregates the RX data streams, originating from all RRHs within the subsystem, via connection (1). The combined RX data stream is sent through P2P streams towards up to 4 MIMO processors using connection (2). The RRH Splitter distributes the combined TX data stream received from the MIMO processors to all other RRHs in the subsystem. In total, there are up to 8 subsystems comprising up to 8 RRHs each. Each subsystem connects to the MIMO processor through the Combiner RRH (1 DMA FIFO) and the Splitter RRH (1 DMA FIFO), that is, two DMA FIFOs are required to connect a subsystem

to a MIMO processor. Thus, a MIMO processor can connect to 8 subsystems using 16 DMA FIFOs, which meets the requirement of 16 DMA FIFOs per MIMO processor.

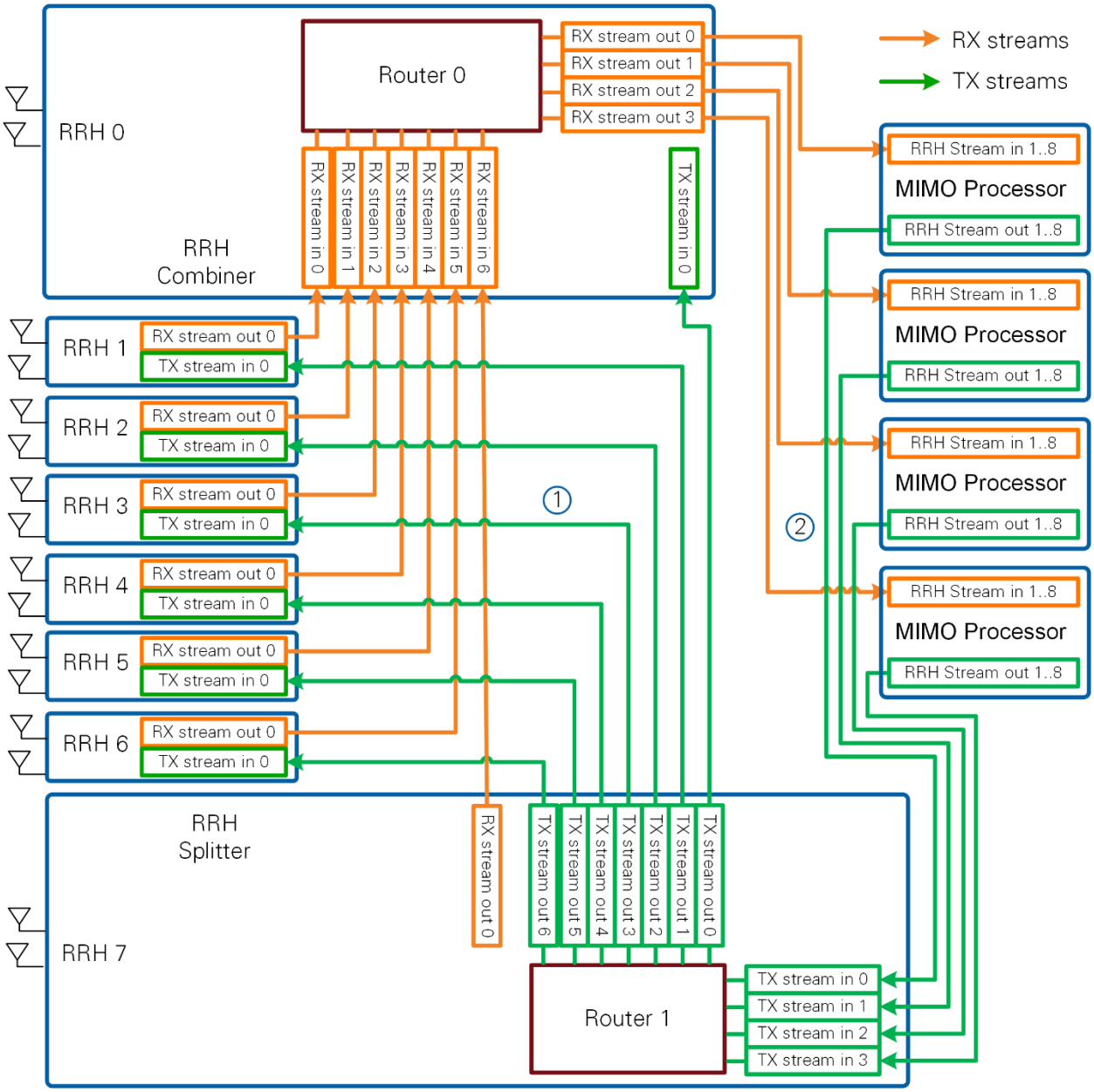

Figure 4-4: Subsystem P2P Connections

4.2.2 Number of MIMO Processors

MIMO processors address two challenges. First, the aggregate inbound and outbound data rate can be very high at large antenna counts. Second, it may be hard to fit the MIMO processing for all subcarriers on a single MIMO processor. These challenges are addressed by using multiple MIMO processors. Each MIMO processor connects to all RRHs but processes only a subset of subcarriers as explained below.

The MIMO processors uses PXIe-7976 modules. They offer a theoretical full duplex 4 Gbyte/s PCIe throughput (PCIe Generation 2 with 8 lanes). They interconnect with the subsystem's combiner and splitter. Those subsystems use USRP devices with a theoretical full duplex data rate of 1 Gb/s (PCIe Generation 1 with 4 lanes). Assuming the full data rate is used on all combiners up to 8 Gb/s would be transferred towards the MIMO processor. This exceeds the capacity of the FlexRIO PCIe endpoint. To save some resources for communication towards Host and Bit Processor the data rate should stay below 2 Gb/s for each FlexRIO module. Since all antenna streams are required for MIMO processing, the actual data rate is based on the number of antennas. Three different configurations, shown in Table 4-1, derive from those considerations. One MIMO processor FPGA module is required per 2 subsystems, for the combined traffic to stay below 2 Gb/s. In the case where more than 2 subsystems are employed, multiple MIMO processor FPGA modules are required and the RRH streams are then split in frequency domain across the different MIMO processors (see section 4.3.1).

Table 4-1: MIMO Processor Configurations

MIMO processors	Max subsystems	Max Antennas M	RBs/MIMO processor
1	2	32	100
2	4	64	50
4	8	128	25

The higher the number of antennas, the more MIMO processors are chosen. Also, the more antennas, the fewer resource blocks are processed per MIMO processor.

4.2.3 Bit Widths

A small number of typical bit width can be found in the FPGA designs. These bit widths have been chosen for the following reasons

- Xilinx DSP48 slices are used to implement high-throughput multipliers. These slices support multiplication of signed 18-bit wide values with signed 25-bit wide values. The concatenation of two DSP48 slices supports multiplication of 25-bit wide values with 35-bit wide values. Thus, the bit width is often chosen as 18-bit, 25-bit or 35-bit.
- In-phase/quadrature (I/Q) samples can be represented at 16-bit per I and Q with minimal distortion in many cases. Also 32-bit I/Q values are easy to package before transmission over inter FPGA P2P interfaces. Alternatively, 12 bit per I and Q are also used, specifically to reduce the data rate on some FPGA P2P interfaces.

4.3 Data Routing

This section explains the data format which has been chosen to exchange data between the different hardware modules.

4.3.1 RRH to MIMO Processor

This subsection explains data rate and data format related to the interface between RRHs and the MIMO process.

Bit width per I/Q symbol: A subsystem comprises up to 8 RRHs. It is connected to a MIMO processor through 1 RRH in Tx (Splitter) and 1 RRH in Rx direction. The capacity on this connection determines the bit width per I/Q symbol as follows.

The USRP devices offer a theoretical full duplex capacity of 1 Gb/s (PCIe Generation 1 with 4 lanes). About 80% of this capacity should be used. The remaining capacity is left for PCIe management traffic and header data to be transmitted.

RRHs and MIMO Processors exchange frequency domain I/Q symbols (for example, QAM symbols) for each antenna over interfaces 1 and 2 as shown in Figure 4-1. The expected throughput based on the LTE radio frame schedule can be calculated as

$$1200\ \frac{\text{I/Q symbols}}{\text{OFDM symbol}} \cdot 140 \frac{\text{OFDM symbols}}{\text{radio frame}} \cdot 100 \frac{\text{radio frames}}{\text{s}} \tag{4.1}$$

$$= 16.8 \cdot 10^6\ \frac{\text{I/Q symbols}}{\text{s}}.$$

To transmit all frequency domain I/Q symbols continuously between (combiner or splitter) RRH and MIMO processor, the maximum bit width of the frequency data follows as

$$2 \frac{10^9 \frac{\text{byte}}{\text{s}} \cdot 8 \frac{\text{bit}}{\text{byte}} \cdot 0.8}{16.8 \cdot 10^6\ \frac{\text{I/Q symbols}}{\text{s}} \cdot 16\ \frac{\text{antennas}}{\text{subsystem}}} = 23.8\ \frac{\text{bits}}{\text{subcarrier}}. \tag{4.2}$$

The application framework therefore uses 24 bits/ I/Q symbol and 12 bits per I and Q on the connection between RRH and MIMO processor.

Packet format at each RRH: Each RRH exchanges transmitted/received frequency domain I/Q symbols with the splitter or router RRH over a P2P connection (Interface 1). To use the P2P protocol efficiently, this data must be aligned to a common byte boundary. For this reason, eight I/Q symbols, corresponding to four consecutive subcarriers at two antennas are packed into three unsigned 64-bit integers as shown in step 1 in Figure 4-5.

Packet format at combiner and splitter RRH: Combiner and splitter RRHs exchange data packets representing one RB corresponding to the number of antennas in the subsystem (16 antennas maximum) with the MIMO processor over interface 2. The respective packet format is shown in step 2 in Figure 4-5. Note the amount of data per RB varies depending on the number of antennas per subsystem.

Routing RBs to multiple MIMO processors: The FPGA resources of a MIMO processor FPGA do not suffice to implement equalization for all subcarriers in real time. Thus, this task is split between up to four MIMO processors depending on the number of antennas (see section 4.2.2). Each MIMO processor processes only a subset of RBs. A combiner RRH distributes complete RBs to MIMO processors as shown in step 3 in Figure 4-5. Likewise, a splitter RRH collects RBs from multiple MIMO processors. One RB is read from or sent to a MIMO processor before switching to the next MIMO processor. This way, the long periods of peak traffic at a single MIMO processor are avoided. The PCIe

switches can serialize the few simultaneous PCIe packets per RB towards one MIMO processor while already serving the next MIMO processor. A MIMO processor can start processing once it has received data corresponding four subcarriers and all antennas in the system (see section 5.2.2 for details).

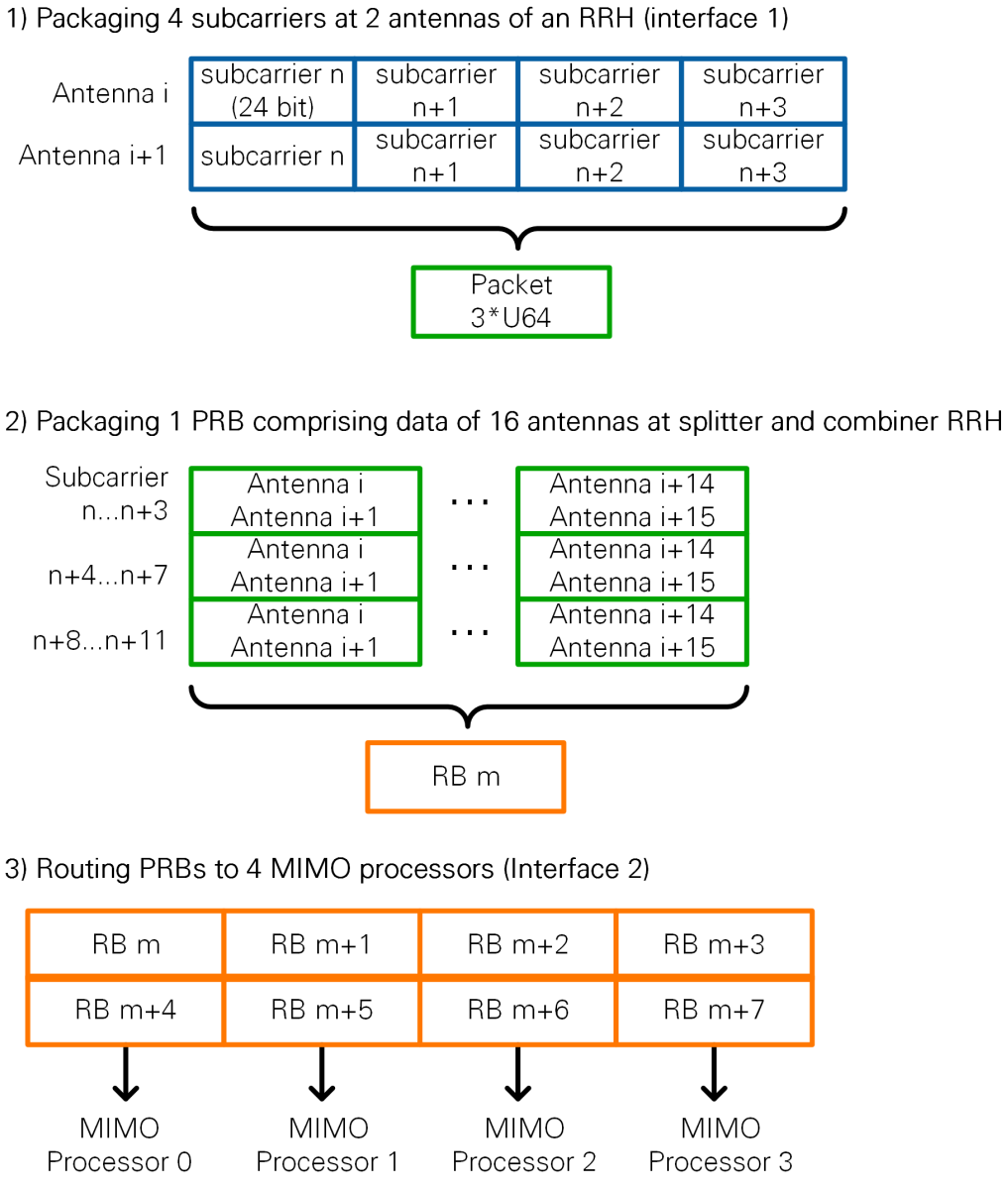

Figure 4-5: Packet Format of RRH and MIMO Processor for a 128-antenna System Comprised of Four MIMO Processors

Note: Data for all OFDM symbols in a radio frame is transmitted between MIMO processor and RRHs. In the transmit chain, OFDM symbols which are allocated for the receive direction, are filled with zero values, and vice versa for the receive direction.

4.3.2 MIMO Processor and Bit Processor

The MIMO and bit processor exchange frequency domain I/Q data symbols corresponding to 12 spatial layers and 1,200 used subcarriers per OFDM symbol over interface 3 as shown in Figure 4-1. The data rate depends on the configured frame schedule, for instance on the number of pilots OFDM symbols. The calculation in (4.3) takes the maximum rate into account (only data OFDM symbols)

$$1200\,\frac{\text{I/Q symbols}}{\text{OFDM symbol}} \cdot 140\,\frac{\text{OFDM symbols}}{\text{radio frame}} \cdot 100\,\frac{\text{radio frames}}{\text{s}} = 16.8 \cdot 10^6\,\frac{\text{I/Q symbols}}{\text{s}}. \tag{4.3}$$

For 12 layers, the I/Q symbol rate in TX or RX direction is determined in the following equation:

$$16.8 \cdot 10^6\ \frac{\text{subcarriers}}{\text{s}} \cdot 12\ \text{layers} = 201.6 \cdot 10^6\ \frac{\text{subcarriers}}{\text{s}} \tag{4.4}$$

On TX and RX path, the application framework uses 16 bits to represent the in-phase and quadrature phase part of an I/Q symbol. This results in a total of 806.4 Mbyte/s in case there is only one MIMO processor. The considerations in section 4.2.2 accounted for additional P2P traffic of the MIMO processors.

Two subcarriers are combined to 64 bit to make use of the same data routing mechanisms used for the RRH-to-MIMO processor P2P interfaces.

The bit processor uses the same RB-wise scheduling to MIMO processors as compared to the RRH P2P interfaces to the MIMO processors.

Based on the output format of the MIMO Processor RX chain the data is ordered as shown in Figure 4-6 (see section 5.2.3 for details).

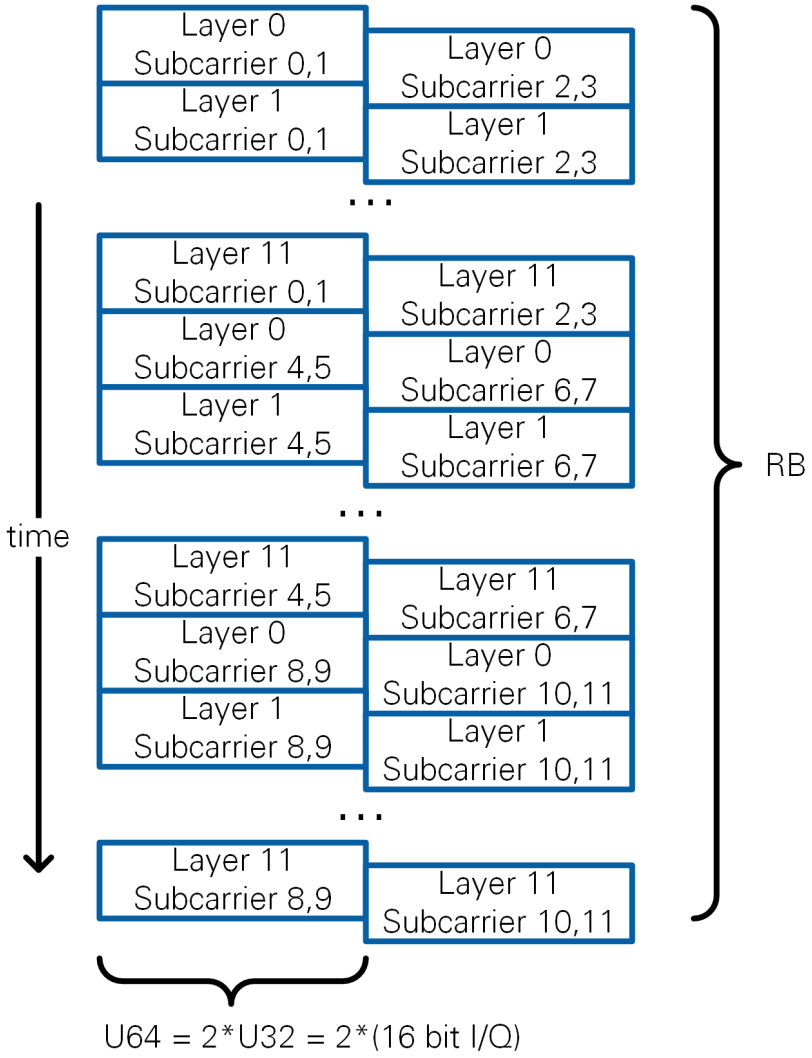

Figure 4-6: Packet Format of MIMO Processor - Bit Processor P2P Interface

4.3.3 Host Traffic

The host exchanges received payload data or data to be transmitted with the bit processor in U8 format over interface 4, as shown in Figure 4-1. See section 2.3 for payload packet formats.

Transmit direction:
DTP packets or PN packets are transmitted over host-to-target DMA FIFOs to the bit processor FPGA. There are 12 DMA FIFOs, one for each data source as described in section 2.3.4. Each data source is uniquely coupled to a mobile station.

Receive direction:
The bit processor FPGA has extracted valid received DTP packets and formatted them into receive indications. These receive indications are forwarded to the host using a single DMA FIFO. The bit processor FPGA multiplexes receive indications corresponding to up to 12 mobile stations into a single DMA FIFO. The host uses the header per receive

indication to de-multiplex these receive indications and assign them to data sinks (UDP ports).

4.3.4 Throttling and Packetizing

The access to P2P FIFOs on the FPGAs has been carefully designed to maximize the throughput and minimize the P2P overhead as follows.

- The P2P implementation operates most efficiently in terms of minimizing the overhead of control and header overhead when data chunks of 512 bytes are transmitted. Thus, there is a local FIFO implemented before each P2P FIFO writer. Once 64 U64 values (512 bytes) are available in the local FIFO and enough space is available in the P2P FIFO, these 64 U64 values are transferred from the local FIFO into the P2P FIFO.
- It is ensured that only 80% of the supported maximal data rate of a PCIe connection is used. This leaves enough margin for PCIe control and header information. This is particularly important for the connection of splitter or combiner RRHs to the MIMO processor or for the P2P connections within RRH subsystems. The data rate reduction is achieved by throttling the access to P2P FIFOs as follows:
    - Routers are implemented in a clock domain of 105 MHz.
    - Transmitting one U64 value per clock cycle results in 840 Mbyte/s.
    - Additional rate limiters throttle the input to P2P FIFOs.
    - The aggregated data rate at combiner RRH input hence cannot exceed 840 Mbyte/s.

    For example, all RRHs, except for the combiner RRH transmit one U64 value per 8 clock cycles, which results in a maximum throughput of 105 Mbyte/s.

5 Base Station FPGA Implementation

This chapter presents base station FPGA implementation details. The focus of this chapter is on the three main FPGA designs which are used in the base station. This chapter does also serve as a reference for the mobile stations. Only the increment or difference in the respective implementation is covered in chapters 6 and 7.

5.1 RRH

5.1.1 Overview

RRHs are implemented on USRP devices. Each RRH implements two radio transceivers and the respective transmit and receive OFDM processing. Groups of eight RRHs form a subsystem as explained in section 4.1.1. RRHs receive precoded subcarriers including pilots from the MIMO processor. RRHs deliver received subcarriers including pilots to the MIMO processor. Router functionality determines which subcarriers are sent to which MIMO processor and which subcarriers are received from which MIMO processor, as described in section 4.3.1.

An overview block diagram of a single RRH is shown in Figure 5-1. Only one of the two radio transceiver chains and the respective signal processing are shown.

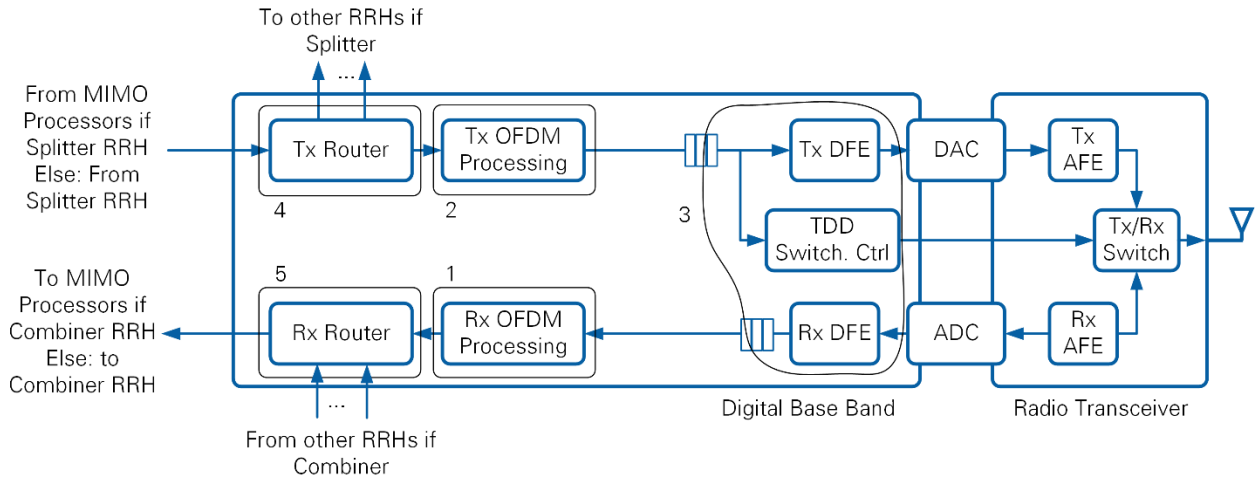

Figure 5-1. Base Station RRH Overview

The digital baseband part comprises the following five main loops:

1. Rx OFDM processing including FFT functionality.
2. Tx OFDM processing including IFFT functionality.
3. A digital frontend. It is connected to the first two loops through FIFOs. It contains the control for time division duplex (TDD) switching which is briefly covered in section 5.1.7. The digital frontend itself is standard USRP functionality and is not further covered.
4. Tx Router functionality which is enabled in case the RRH acts as a splitter.
5. Rx Router functionality which is enabled in case the RRH acts as a combiner.

The same RRH implementation is shared between the multi-antenna MS, the BS RRH combiner and BS RRH splitter. Parts of the code are disabled or unused, depending on personality.

5.1.2 Inter-RRH Time and Frequency Synchronization

All RRHs of the base station are time and clock synchronized. This is achieved through shared trigger and 10 MHz clock signals (see section 4.1.1).

The synchronous transmit timing is achieved as follows:

1) The host starts the transmit data generation on the bit processor FPGA. Accordingly, FIFOs as part of the transmit data path get filled.
2) The host checks whether the transmit chain of all RRHs has been filled with transmit data.
3) Once this is the case, the host sends a trigger to RRH0. RRH0 forwards this trigger to the timing module, which further forwards it to the different CDA-2990 devices for distribution to all RRHs.

4) All RRHs including RRH0 receive this trigger synchronously. All RRHs start the transmission simultaneously.

The radio frame start of the receiver is derived from the radio frame start at the transmitter plus a delay. The following section explains the transmit – receive timing calibration.

5.1.3 Transmit-Receive Timing Calibration

Figure 5-2 shows the block diagram of the digital front end (DFE) and radio transceiver (RF) part of the RRH.

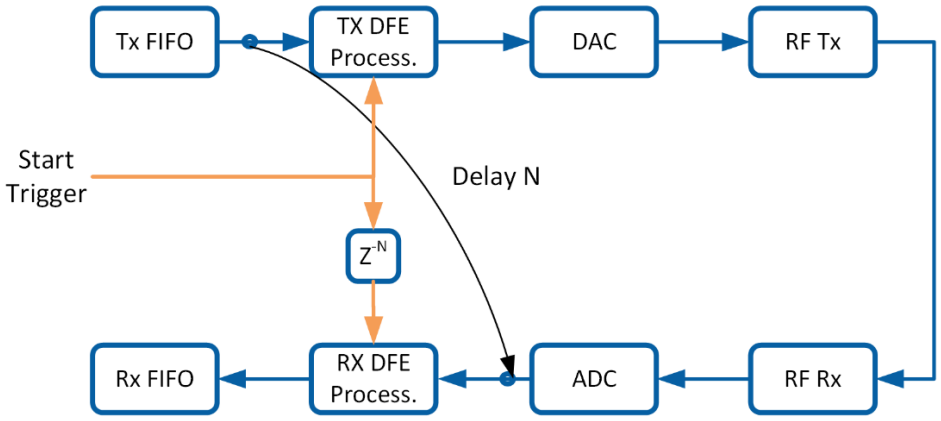

Figure 5-2. Transmit-Receive Timing Calibration at the BS RRH

Transmit data is fed to the DFE loop through a FIFO (Tx FIFO). Likewise, the received signal is shared with the receive OFDM processing loop through a FIFO (Rx FIFO). Assume the transmit and receive chain are configured in a loopback mode as shown in Figure 5-2. The first sample which is read from the Tx FIFO is received at the input to the Receive DFE processing with a delay. Purpose of the calibration is to determine this delay. The delay is used to start the receive DFE later as compared to the transmit DFE. In consequence, the first valid sample ever read from the Tx FIFO would be the first valid sample ever transferred into the Rx FIFO. Note that the start trigger is received at all RRHs at the same time.

The delay is determined by configuring the transmit and receive chain in loopback mode as shown in Figure 5-2. A signal is read from the Tx FIFO and transmitted. Received values are written into the Rx FIFO without delay ($N = 0$). This received sequence is evaluated offline. The transmitted signal can be found N samples after the start of the received sequence.

5.1.4 Receive Path OFDM Processing

Figure 5-3. shows the receive OFDM processing for two antennas. Grey parts are synchronization related. They are not active at the BS RRH. They are used at the multi-antenna mobile station as described in section 7.1.

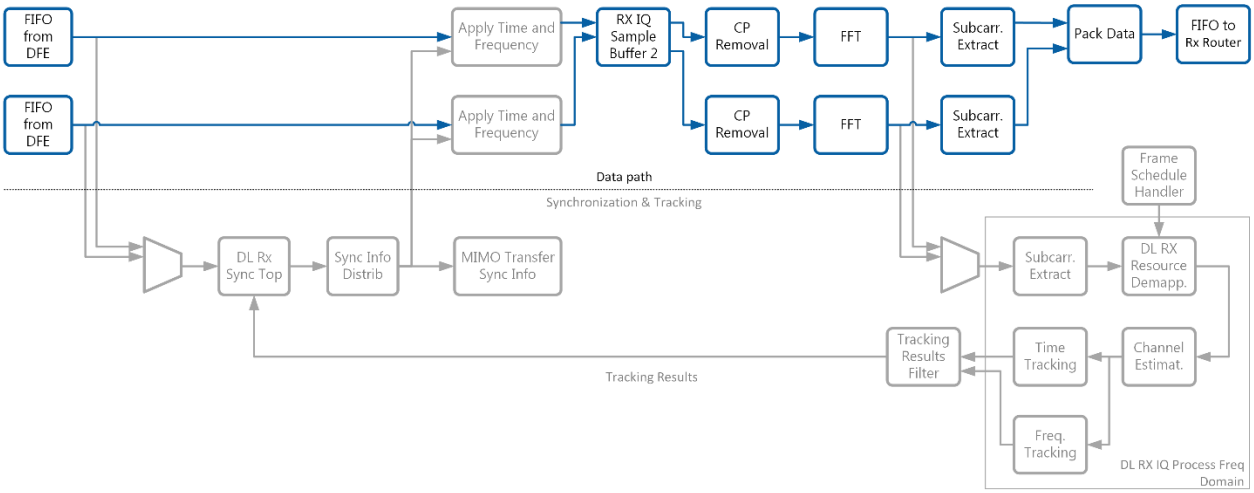

Figure 5-3. RRH Receive OFDM Processing

The processing is implemented as follows:

- The received time domain signal at 30.72 MHz is fed into the **RX IQ Sample Buffer 2** block. This block buffers incoming samples, aligns them with an OFDM symbols start signal and a sample valid signal, and delivers consecutive OFDM symbols to subsequent blocks.
- The **CP Removal block** removes the valid flag from all CP samples in a radio frame.
- The **FFT block** converts 2,048 time domain samples to frequency domain using a Xilinx FFT [11]. A continuous phase shift between samples by 180 degree (toggling negation) corresponds to a cyclic FFT shift and shifts the DC carrier to the 1024th output sample of the FFT. That is, the middle carrier corresponds to the DC carrier.
- The **Subcarrier Extract block** provides the 1200 used subcarriers per OFDM symbol at its output.
- The frequency domain subcarriers corresponding to two antennas are packaged by the **Pack Data** block as described in section 4.3.1 and forwarded to the Rx router (combiner).

5.1.5 Transmit Path OFDM Processing

Figure 5-4 shows the transmit OFDM processing for two antennas. Grey parts are synchronization related. They are not active at the BS RRH. They are used at the multi-antenna mobile station.

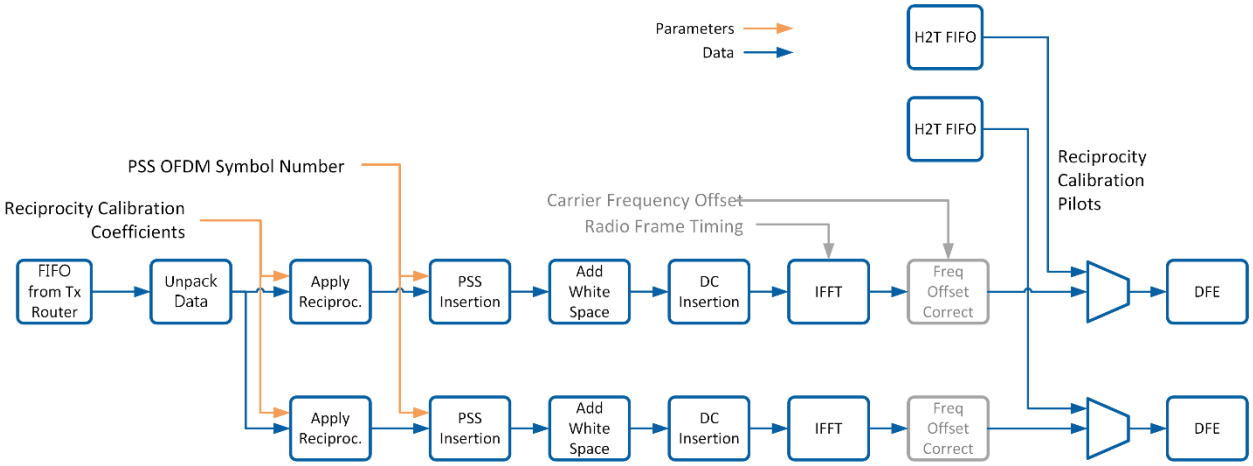

Figure 5-4. RRH transmit OFDM Processing

The processing is implemented as follows:

- Used subcarriers are received from the Tx Router (originating from the MIMO processor) in the packet format described in section 4.3.1.
- The **Unpack Data block** extracts received subcarriers (1,200 used subcarriers) for both receive antennas from the packets provided by the MIMO processor.
- The **Apply Reciprocity block** receives reciprocity calibration coefficients for each subcarrier from the host. These coefficients are stored in a memory on the FPGA. Each subcarrier is multiplied by the corresponding reciprocity calibration coefficient as described in section 3.6.5.
- The **PSS Insertion block** reads the frequency-domain PSS from a memory and inserts it into at the correct OFDM symbol position. This position is configured from the host based on the radio frame schedule. Note that the data provided to the PSS insertion block comprises valid place holder data where the PSS is placed. This is replaced by the PSS sequence.
- The **Add White Space block** adds 424 and 423 zero subcarriers to the edges of the used 1200 subcarriers. The **DC Insertion block** inserts the zero DC subcarrier. The output of this block comprises ,2048 valid subcarriers.
- The **IFFT block** computes the time domain wave form using a length 2,048 XILINX IFFT [11]. In addition, a CP is prepended. This block implements the option to adapt the transmit timing by varying the length of the CP. This functionality remains inactive at the base station. The IFFT operation is followed by inverting the sign of every second time domain symbol. This operation implements an FFT shift.
- The **Frequency Offset Correction block** remains unused at the base station.
- A multiplexer allows to transmit time domain sequences provided by the host. This functionality is used to transmit pilot sequences during the reciprocity calibration procedure which is executed during the initialization of the system.
- The time domain samples are forwarded to the digital frontend, which implements sample rate conversion, for instance.

Note that the transmit I/Q processing generates a transmit signal for all OFDM symbols in a radio frame and for receive OFDM symbols. The signal for these receive OFDM symbols is set to zero and sent to the DFE loop as well.

5.1.6 Router

Each RRH implements two routers which can act as a splitter (Tx Router) or as a combiner (Rx Router). The transmit OFDM signal processing path is sourced from the Tx Router. The receive OFDM signal processing path feeds the Rx Router.

A Router can route data from eight source FIFOs to eight target FIFOs. Figure 5-5 shows a simplified block diagram of the router. Note that pipeline stages, which are part of the design, are not shown.

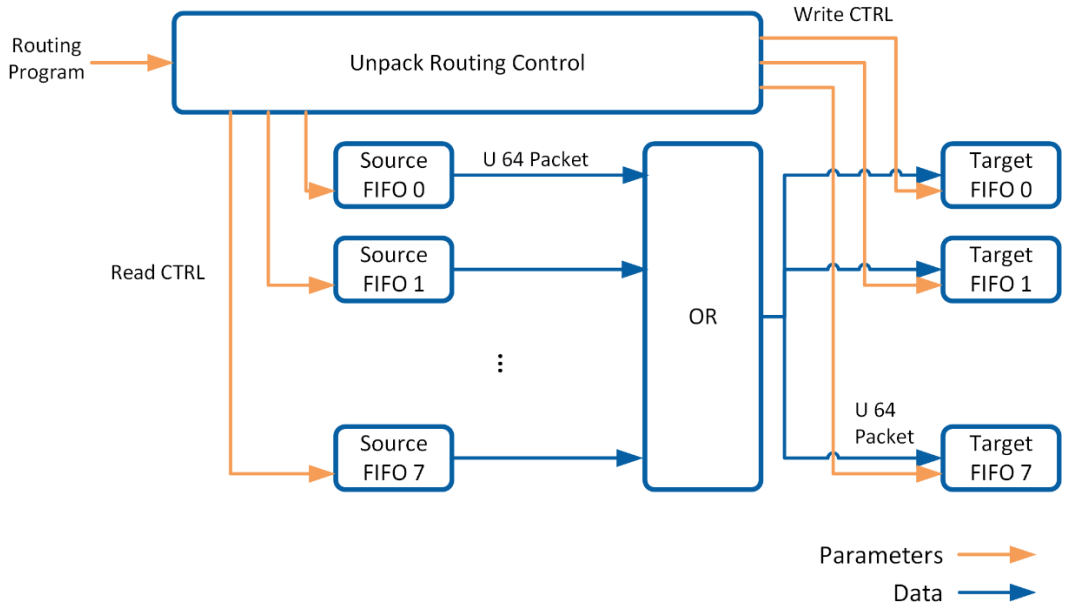

Figure 5-5.RRH 8x8 Router

The operation of the router is as follows:

- The router transfers U64 words from a source FIFO to a target FIFO.
- Source and destination are controlled through a router program. The router program determines source and destination. The same router program is used for all OFDM symbols in a radio frame.
- The router program is computed at the host and transferred to the FPGA (Unpack Routing Control block) on initialization time. It is represented in terms of a length-8,192 sequence. Each U8 word in the sequence encodes source and destination FIFO index in 3 bits, respectively. One element of the length-8,192 sequence is used to route one U64 word. Routing is implemented by enabling the source and destination FIFO corresponding to the 3-bit indices. Note that 3 consecutive U64 words correspond to eight I/Q symbols (four subcarriers at two antennas). Refer to section 4.3.1. for more details.

- A U64 word is routed if a word is available in the source FIFO and there is space available in the target FIFO.

The router is configured in different ways, depending on the RRH. Note that RRH0 acts as a combiner, which collects received data from all RRHs in a subsystem and forwards it to the MIMO processor. The last RRH (RRH7 if there are eight RRHs per subsystem) acts as a splitter which distributes data originating from the MIMO processor to all other RRHs in the subsystem. All other RRHs act as normal RRHs. The router configuration is summarized in Table 5-1.

Table 5-1. Router Configurations

RRH Type	Tx Router	Rx Router
Last RRH in the subsystem – Splitter Personality	Four source FIFOs which connect to the MIMO processor and one debug FIFO are routed to eight target FIFOs. One target FIFO is a local FIFO that is connected to the local OFDM transmit processing chain, the remaining seven FIFOs connect to other RRHs in the subsystem.	One source FIFO is routed to two target FIFOs. One target FIFO connects to the combiner RRH7. The other target FIFO is a debug FIFO.
RRH0 – Combiner Personality	Two source FIFOs connect to one target FIFO. One source FIFO connects to RRH0 (splitter). The other source FIFO is a debug FIFO.	Eight source FIFOs connect to five target FIFOs. One source FIFO is a local FIFO that is connected to the local OFDM receive processing chain. The remaining seven FIFOs connect to other RRHs in the subsystem. Four target FIFOs connect to the MIMO processor. One target FIFO is a debug FIFO.
Normal RRHs	Two source FIFOs connect to one target FIFO. One source FIFO connects to RRH0 (splitter). The other source FIFO is a debug FIFO.	One source FIFO is routed to two target FIFOs. One target FIFO connects to the Combiner RRH7. The other target FIFO is a debug FIFO.

Note that debug FIFOs mentioned in Table 5-1 are not exposed in the base station or mobile station host code.

5.1.7 TDD Switching

TDD switching control is implemented as part of the DFE loop as shown in Figure 5-1. Note that this functionality is implemented identically at the BS and the MSs. The DFE

loop receives a trigger which indicates the start of a radio frame. Also, the DFE loop is aware of the radio frame schedule.

A counter in the DFE loop counts the transmitted samples received from the transmit I/Q processing loop and determines the start of an OFDM symbol based on this count. Also, it determines the OFDM symbol type (uplink/downlink) by comparing the OFDM symbol index with the radio frame schedule. A Boolean value is set to true, whenever the OFDM symbol type corresponds to a transmit symbol. This Boolean value is transferred into a register which controls the transmit receive switching which is part of the radio transceiver chain.

5.2 MIMO Processor

5.2.1 Overview

MIMO processors aggregate signals from all antennas as shown in Figure 4-4.
On the RX path, the frequency domain I/Q symbols are received from the combiner RRHs. Channel estimation and MIMO equalization are performed. The resulting equalized frequency domain I/Q symbols are forwarded to the bit processor.

On the TX path, modulated I/Q symbols are received from the bit processor, pilots are added and precoding is applied. The precoding is based on the MIMO equalizer matrices computed at the receiver.

The MIMO processor can be used for the base stations and multi-antenna mobile station. A Boolean switch allows to switch TX and RX path to be active for DL and UL or the other way around. In the following sections the MIMO processor is assumed to work in the base station context, where the received signals correspond to the uplink and the generated signals correspond to the downlink.

The MIMO Processor design is the same for all configurations in Table 4-1. A compile-time switch enables one of the configurations. The configuration is an enumeration type on the FPGA which selects one configuration for each row in Table 4-1. All other parameters are derived from this enumeration input provided to most of the modules.

5.2.2 FPGA Considerations

The following subsections explain how data width and clock rates of the data path are derived from the system design. The RX chain serves as a reference for the calculation. The results are also valid for the TX chain.

5.2.2.1 Parallelism and Clock Rate.

The MIMO processor inbound/outbound rate at the interface to the RRH is static regardless if 128 or 32 antennas are used. Four MIMO processors process a quarter of the RBs per OFDM symbol each for 128 antennas. A single MIMO processor processes all RBs for 128/4=32 antennas. This rate computes as shown below

$$32 \cdot 12 \frac{\text{I/Q symbols}}{\text{RB}} \cdot 100 \frac{\text{RB}}{\text{OFDM symbol}} \cdot 140 \frac{\text{OFDM symbols}}{\text{radio frame}} \cdot 100 \frac{\text{radio frames}}{\text{s}} = 537.6 \cdot 10^6 \frac{\text{I/Q symbols}}{\text{s}}. \quad (5.1)$$

Serial processing of I/Q symbols would require at least a clock rate of 537.6 MHz, which is not desirable. By processing four I/Q symbols in parallel and lowering the clock rate to 200 MHz, we can process $800 \cdot 10^6$ I/Q symbols per s.

A clock rate of 200 MHz also guarantees that channel estimation and computation of the MIMO equalizer matrix take less time than the duration of an OFDM symbol. Thus, the data OFDM symbol, which follows a pilot OFDM symbol, can already be processed using updated equalizer matrices, computed based on the pilot symbol.

5.2.2.2 Matrix Transfers

The MIMO processor mostly operates on matrices. The matrix dimensions would typically depend on the number of antennas and the number of spatial layers. The design of the MIMO processor and all processing steps, however, are dimensioned for a static maximum number of antennas and spatial layers. The MIMO processor design and dimensioning is determined on compile time. That is, there is one dedicated bit file for each configuration in Table 4-1.

Figure 5-6 presents a simplified high-level view on the MIMO processor processing steps and its interfaces to RRHs and the bit processor.

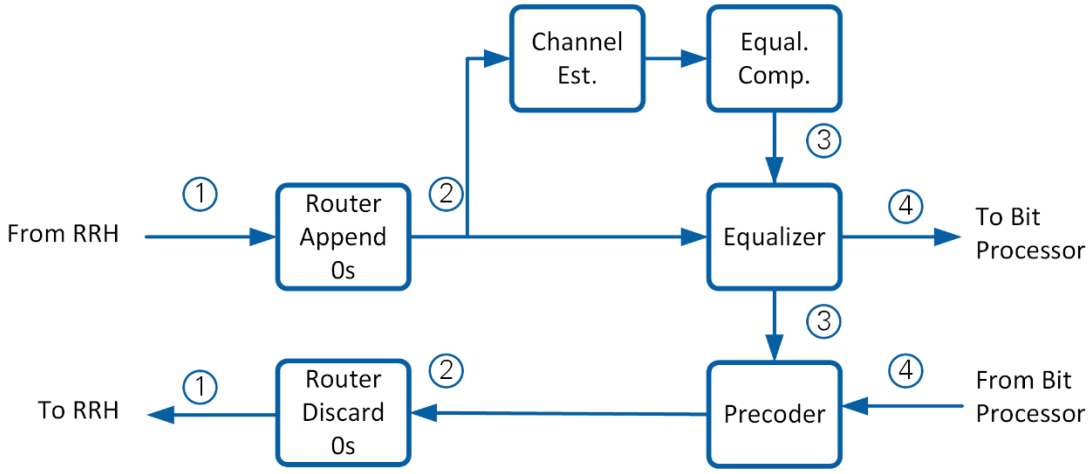

Figure 5-6. Processing Steps on the MIMO Processor

Numbers indicate matrices which are introduced in Figure 5-7.

Data is communicated with RRHs using **matrix (1)**. Matrix (1) contains precoded or received I/Q symbols corresponding to all base station antennas and one RB (12 subcarriers).

A MIMO processor may be designed for 32 antennas while the actual number of antennas is only 16. Matrix (1) would have 16 rows and 12 columns in this case.

To implement a static processing for 32 antennas, the router blocks append/discard some rows containing zero values to matrix (1) to generate **matrix (2)** of static dimensions. In the example, 16 rows containing zeros would be appended. Matrix (2) would have 32 rows and 12 columns. Matrix (2) is passed to channel estimation, equalization and precoding.

Channel estimation equalizer computation results in **matrix (3)** which contains the weights used for equalization and precoding. Matrix (3) contains some columns with zero values. The number of columns corresponds to the difference between the maximum number of antennas and the actual number of antennas. In the example, 16 columns would contain weights and 16 columns would contain zero values.

The equalizer equalizes the received data comprised in matrix (2) using the weight matrix (3). The result is **matrix (4)** containing I/Q symbols which correspond to 12 layers and 12 subcarriers (1 RB). The dimension of matrix (4) is static, regardless of the actual number of layers which is enabled. Matrix (4) is exchanged with the bit processor FPGA.

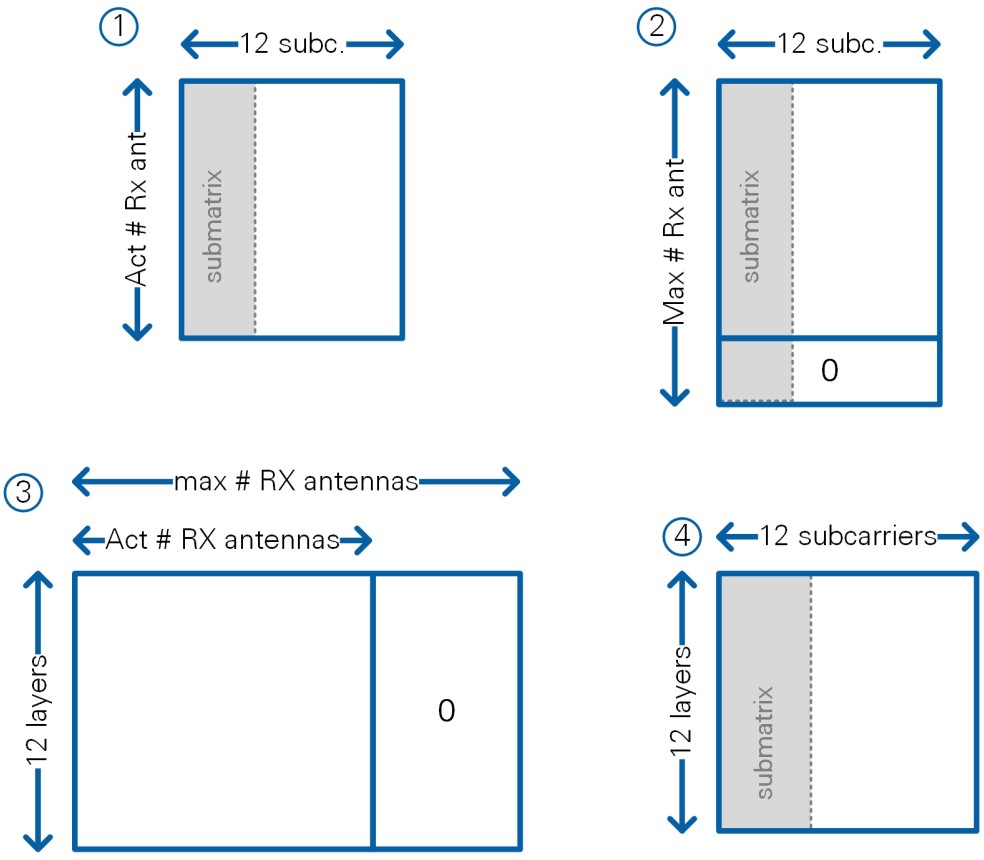

Figure 5-7: MIMO Processor Matrices

Note that the submatrices in matrices (1), (2), and (4) represent four consecutive subcarriers. These four subcarriers are processed in parallel. Matrices (1), (2), and (4)

comprise three of these submatrices that represent an RB and are processed sequentially. Per clock cycle, a submatrix row comprising four I/Q symbols is transferred between modules. Regarding weight matrix (3) a complete column with 12 elements is transferred per clock cycle.

The MIMO processor design ensures that complete matrices (1),…, (3) are passed from one module to another without any idle clock cycles. This data pattern is known and exploited in all modules. For example, any operations can follow a static schedule after having received a start flag.

5.2.2.3 Interface to RRH FPGAs

I/Q symbols are received in packets, comprising eight I/Q symbols per packet. Each packet spans three U64 words (see section 4.3.1). Thus, it takes three clock cycles to read eight I/Q symbols. As explained in the previous subsection the data path must operate on four subcarriers in parallel. To provide four I/Q symbols per clock cycle, the U64 words would have to be read at a clock rate which is 1.5 times higher as compared to the processing clock rate. Three cycles of reading at the faster clock rate would then allow two cycles of processing at the lower clock rate.

Especially for DMA FIFOs, NI recommends using lower clock rates to allow the synthesis tools to handle the connection of all DMA FIFOs towards the PCI express block on the FPGA. To achieve a lower clock rate, the P2P streams are read and unpacked in two parallel chains using two routers. A combiner module merges the output of the two routers into the format of matrix (2). Note that the combiner delivers the three submatrices comprised in matrix (2) sequentially. It delivers each submatrix row by row, that is, four I/Q symbols per clock cycle. A submatrix at the output of the combiner module is a matrix of dimension *maximum number of antennas* x *4 subcarriers*.

The required clock rate for each of the two routers is computed in the following equation:

$$32 \cdot 12 \frac{\text{I/Q symbols}}{\text{RB}} \cdot \frac{3\ \text{U64}}{8\ \text{subcarriers}} \cdot \frac{1}{2\ \text{routers}} \cdot 100 \frac{\text{RB}}{\text{OFDM symbol}} \cdot 140 \frac{\text{OFDM symbols}}{\text{radio frame}} \cdot 100 \frac{\text{radio frames}}{\text{s}} = 100.8 \cdot 10^6 \frac{\text{U64}}{\text{s} \cdot \text{router}}. \quad (5.2)$$

It is sufficient to operate the two routers at a clock rate of 105 MHz.

5.2.3 Receive Path

The MIMO processor receive path, shown in Figure 5-8, combines the P2P source, channel estimation, channel equalization and P2P sink.

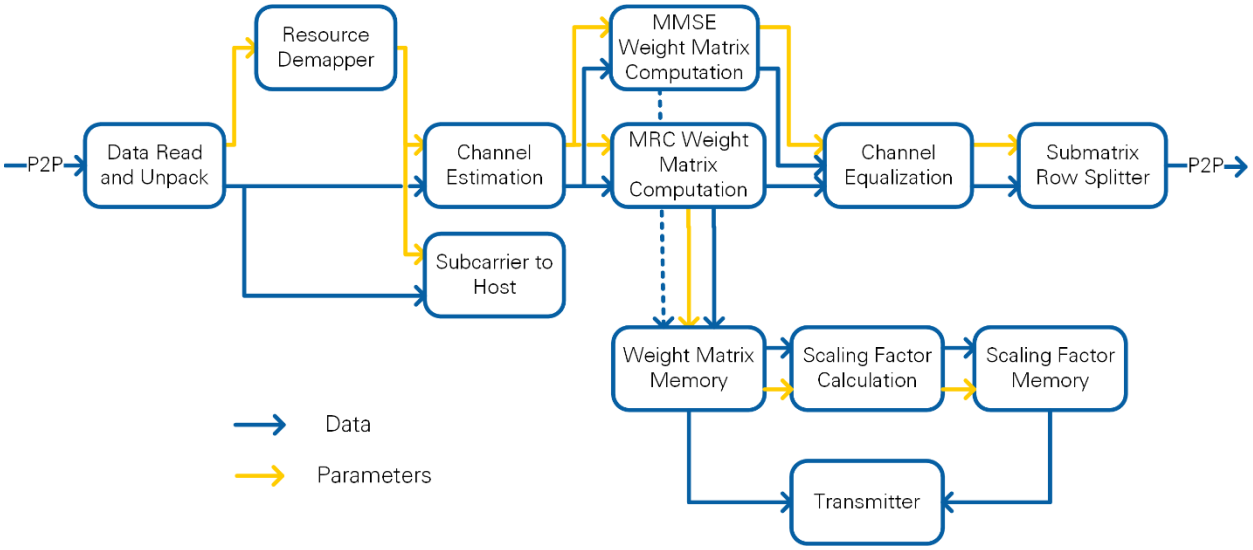

Figure 5-8: MIMO Processor Receive Path

The throughput of all modules is designed to keep up with the throughput of upstream modules. Therefore, no back-pressuring or handshaking is implemented.

5.2.3.1 Data Read and Unpack

The same data router as on the RRH (see section 5.1.6) is used to merge the streams from different RRH subsystems in each data source. The Data Read and Unpack module accounts for the processing periods of the downstream modules. It grants enough clock cycles per RB such that all downstream modules can process each RB.

To split the incoming data traffic in half (as explained in section 5.2.2.3) only half of the RRH subsystems are handled by one router. Based on the antenna configurations the P2P streams are setup as shown in Table 5-2.

Table 5-2: MIMO Processor Subsystem P2P Links

Max antenna	Subsystems router 0	Subsystems router 1
32	0	1
64	0, 1	2, 3
128	0, 1, 2, 3	4, 5, 6, 7

The routing program, executed in each router, is based on the number of connected subsystems and RRHs per subsystem. From each subsystem, packets are read until all antennas have provided four subcarriers (one packet per antenna pair, see first packet in Figure 4-5). Afterwards potential other subsystems are handled in the same way. If the systems' configured antenna count is below the maximum supported number, the remaining non-existing packets are replaced by zeros. For this purpose, there are two FIFOs kept filled with zeros.

5.2.3.2 Subcarrier-to-Host

The subcarrier-to-host block allows to send two OFDM symbols corresponding to all antennas to the host. The OFDM symbol index is user defined. These OFDM symbols typically comprise a pilot OFDM symbol and a data OFDM symbol for host evaluation purposes, such as channel estimation and equalization.

Each I/Q symbol is represented by a complex fixed point number in 2.14 format. It is packed into a U32 word for transmission to the host. The data order is as follows:

- The serial stream sent to the host comprises one complete OFDM symbol after the other.
- The transmission starts with all OFDM symbols corresponding to all antennas for the first requested OFDM symbol index. All OFDM symbols corresponding to the second OFDM symbol index follow in the same order.

Data is only sent if there is space in the respective FIFO to the host. A state machine ensures that all OFDM symbols corresponding to the requested two OFDM symbol indices are sent to the host before new OFDM symbols are being re-ordered at the FPGA.

5.2.3.3 Resource Demapper

The Resource Demapper counts the incoming I/Q symbols and aligns control clusters with these I/Q symbols. A first control cluster contains the OFDM symbol index, the RB index and the OFDM symbol type, derived from the frame schedule. This cluster is intended for filtering of data by downstream modules. The second control cluster contains Boolean flags indicating valid data points. It also marks submatrix, RB, OFDM symbol and radio frame start. All processing blocks align their operation to those flags.

5.2.3.4 Channel Estimation

The channel estimation divides each received subcarrier by the expected pilot sequence (see section 3.3). It uses the fact that the QPSK modulated pilots have an amplitude of 1. Therefore, the division reduces to a multiplication with the conjugate complex pilot sequence. The result of the channel estimation is the MIMO channel matrix shown in Figure 5-9.

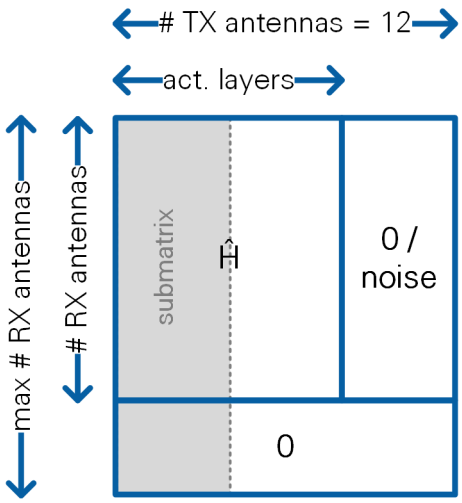

Figure 5-9. MIMO Channel Matrix

The last rows of this matrix contain zeros if the number of receive antennas is less than the maximum number of receive antennas the MIMO processor is designed for. This is enforced by the data input format to the channel estimation module (see matrix (2) definition in section 5.2.2.2).

The channel estimation provides an option to mute the received pilot signal layer-wise by multiplying it with zero. For instance, received pilot signals coupled to layers that are not active (and were no pilot has been transmitted) are set to be zero. This is required for inactive layers to suppress noise, which would lead to numeric instabilities in the equalization calculations. This operation corresponds to columns in Figure 5-9 which contain zero values (or noise in case unused layers are not muted).

Note that the MIMO channel matrix consists of three submatrices of four columns each. The first submatrix is passed to the following module row-by row, followed by the second submatrix, followed by the third submatrix.

5.2.3.5 MMSE and ZF Weight Matrix Computation

The MMSE weight matrix computation is provided by the fix-size MIMO channel estimate, according to Figure 5-9 at its input. The implementation of the MMSE weight matrix computation (see sections 3.4.2 and 3.4.3) is divided into three blocks. First, the extended channel matrix $\boldsymbol{B}$ is derived from the channel estimate in Figure 5-9 by appending a scaled identity matrix in the **Identity Matrix Attachment block**. The result is the extended MIMO channel matrix shown in Figure 5-10.

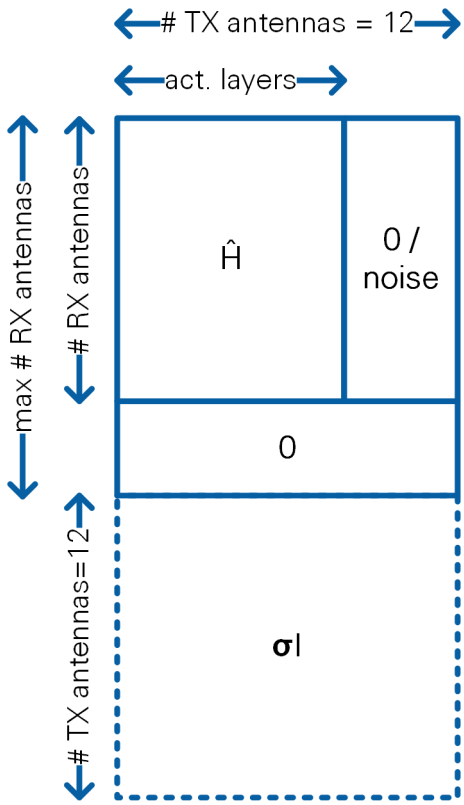

Figure 5-10. Extended MIMO Channel Matrix B

The next step is the calculation of the **QR decomposition** of $\boldsymbol{B}$ followed by the calculation of the MIMO equalizer matrix $\boldsymbol{W}_{MMSE}$ based on the $\boldsymbol{Q}$ matrix within the **Calculate Matrix Inverse block**. The implementation of these modules is described in the following subsections.

5.2.3.5.1 Identity Matrix Attachment

The first part of computing the MMSE equalizer matrix as described in section 3.4.3, is to attach an identity matrix, scaled by σ, to the MIMO channel matrix shown in Figure 5-9. The result is matrix $\boldsymbol{B}$ in Figure 5-10. The identity attachment block receives the first submatrix first, row-by-row, followed by the second submatrix, followed by the third submatrix. After each submatrix 12 rows of four elements each of the identity matrix are appended. Note that the number of valid (channel matrix) samples increases by appending the identity matrix. Dynamic shift registers are used to delay the incoming submatrix rows. The length of the shift registers is increased with every incoming submatrix to compensate for the 12 inserted rows from the identity matrix.

5.2.3.5.2 QR Decomposition

The main part of the MMSE algorithm is the QR decomposition. It uses the modified Gram-Schmidt algorithm to determine the $\boldsymbol{Q}$ matrix. The $\boldsymbol{R}$ matrix is not calculated directly, as only the matrix $\boldsymbol{Q}$ is needed to compute the MMSE matrix inverse as shown in section 3.4.3.

QR decomposition based on the Modified Gram Schmidt algorithm:
The algorithm is described in (5.3). $\boldsymbol{q}_i$ in (5.3) denotes the i-th column of $\boldsymbol{Q}$.

$$\begin{aligned}
&\boldsymbol{Q} = \boldsymbol{B} && \text{\% algorithm operates on } \boldsymbol{B} \text{ directly} \\
&\textbf{for columns } i \textbf{ do} \\
&\quad \boldsymbol{q}_i = \frac{\boldsymbol{q}_i}{||\boldsymbol{q}_i||} \\
&\quad \textbf{for columns } \boldsymbol{k} > \boldsymbol{i} \textbf{ do} \\
&\quad\quad \boldsymbol{q}_k = \boldsymbol{q}_k - \boldsymbol{q}_k^T \boldsymbol{q}_i \boldsymbol{q}_i \\
&\quad \textbf{end for} \\
&\textbf{end for}
\end{aligned} \qquad (5.3)$$

Each column is first normalized and then projected on all remaining columns of the matrix. Normally there would be a division during the projection by the inner product of $\boldsymbol{q}_i$, but due to the normalization in the first line in (5.3), this norm already equals 1 and the normalization step can be skipped.

Parallel Implementation:
This QR decomposition algorithm is implemented in parallel [6] through four core modules: Vector Normalization (N), Vector Projection (P), a repetition module (R) and a module which combines the results (M). See Figure 5-11 for a block diagram of the implementation. Each of the four core modules reads the vector $\boldsymbol{q_i}$ elementwise. The vector normalization module N accumulates the squared absolute value of each incoming element of $\boldsymbol{q_i}$. The original element is delayed until the last element was accumulated into the sum and the reciprocal square root $1/||\boldsymbol{q}_i||$ has been computed. Then, each delayed element in $\boldsymbol{q}_i$ is multiplied with the reciprocal square root. At the same time, a new vector $\boldsymbol{q}_j$ can be provided to the module. The vector projection module P is working similarly. The inner product of the two vector $\boldsymbol{q}_k$ and $\boldsymbol{q}_i$ is derived in this case until the subtraction can start.

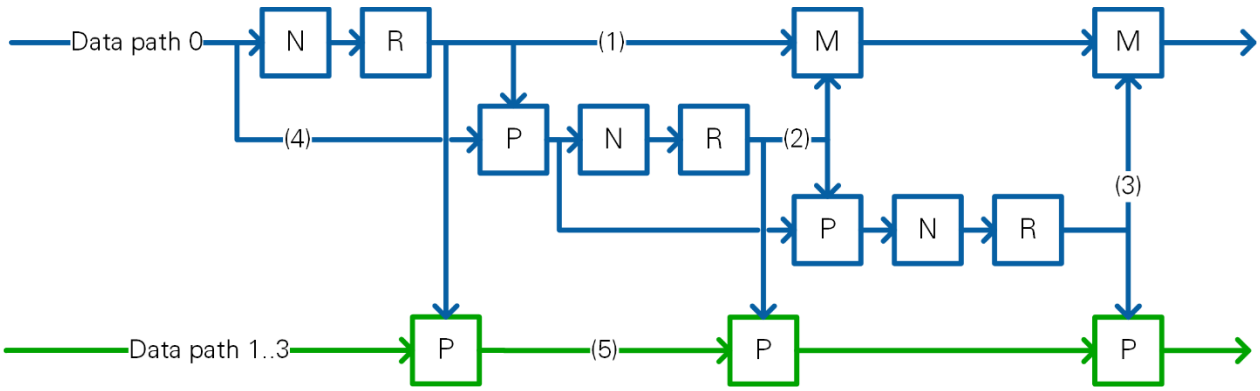

Figure 5-11: QR Decomposition Block Diagram for Single Data Path

For a single data path, the core modules are connected as shown in Figure 5-11. A data path carries one column of submatrices. Additional delays are left out in this figure for simplicity. For each submatrix, one vector of matrix $\boldsymbol{B}$ is provided. For data path 0 the columns 0, 4 and 8 arrive consecutively without waiting cycles. Column 0 is first

normalized in the first N module and provided to connection (1) via the repetition module. Since the vector normalization has a latency, which is around one vector length clock cycles, the next column will be provided in parallel on the QR decomposition input (4). Both vectors can be fed into the first projection module (P). The result is again normalized and provided to connection (2). During the normalization of the second column the third column is processed in the same projection module. Since the first vector is required for this projection as well, the repetition module repeats the normalized first column on connection (1) until new data is provided on the input (4). The result of the second normalization and third provided vector are available at the same time at the second blue P module connected to (2). The result is a combination of all columns provided on data path 0. It is normalized again and available on connection (3). Connections (1-3) must be serialized to provide the output in the same order as on the input. The merge modules M are taking this role. Internally they just select which of the inputs is forwarded to the output. The result on connection (2) has a delay of about twice the vector length clock cycles compared to connection (1). Additional delays are necessary to ensure the second column is available after the last element of the first column has been provided to M. The same applies for the inputs of the second blue M module.

The results computed in the blue part in Figure 5-11 are used to compute the columns 0, 4 and 8 of the Q matrix as they are the first column of the submatrices. The remaining columns of each submatrix are preprocessed in parallel in the green modules. They make use of the repeated normalized columns from connections (1-3) to project those on the other columns. Additional delays must be added to align connection (5) with the parallel nodes on the blue path.

The whole QR decomposition is implemented as shown in Figure 5-12. The normalization chain (NC) contains all blue blocks from Figure 5-11. The projection chain (PC) contains the green blocks. Furthermore, there are block RAM (BRAM) delays (B). They use BRAM memory on the FPGA to delay the data for a longer period. The delay of each B module depends on the remaining operations on the other data paths. Due to the alignment, the output pattern equals the input pattern where the data paths are aligned among each other to form a row of a submatrix.

Due to the data pattern the QR decomposition is slightly modified compared to equation (5.3). The column order of the FPGA implementation is: [0, 4, 8, 1, 5, 9, 2, 6, 10, 3, 7, 11]. The indices i and k in (5.3). Therefore, we use this column order for iterating through the matrix.

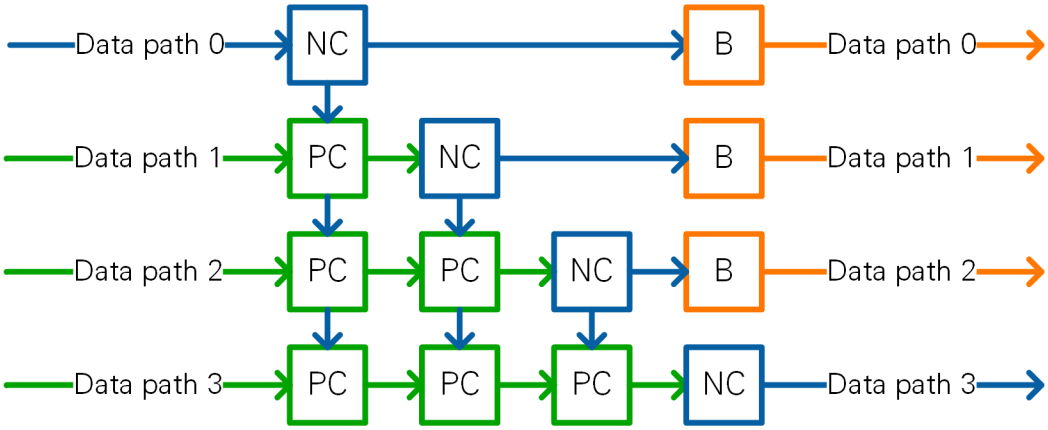

Figure 5-12: QR Decomposition Block Diagram for the Four Data Paths

The QR decomposition is flexible in terms of the vector length which can be configured at compile time. All modules of the QR decomposition use the vector length as input. Based on this number all dynamic delay nodes become fixed sized shift registers during synthesis because of the constant value. The internal bit widths are designed for up to 140 rows in matrix B.

The $\boldsymbol{Q}$ matrix is split into three submatrices, each comprising four columns of $\boldsymbol{Q}$. The first submatrix is passed to the following module row-by row, followed by the second submatrix, followed by the third submatrix.

5.2.3.5.3 Matrix Inversion

The Calculate Matrix Inverse module calculates the matrix $\boldsymbol{W}_{MMSE}$ from $\boldsymbol{W}_{MMSE} = \frac{1}{\sigma}\boldsymbol{Q}_2\boldsymbol{Q}_1^H$ (see (3.17)). The input matrix $\boldsymbol{Q}$ is split into the two matrices $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$. Fixed-length shift registers take care that the rows of matrix $\boldsymbol{Q}_1$ are delayed until all rows of $\boldsymbol{Q}_2$ arrived. The $\boldsymbol{Q}$ matrix is transferred in terms of 3 submatrices into the matrix inversion block as explained in the previous subsection. The matrix product $\boldsymbol{Q}_2\boldsymbol{Q}_1^H$ is hence computed in three steps. Each step, shown in different colors in Figure 5-13, is executed sequentially for one submatrix of $\boldsymbol{Q}$. The result of the first four columns (indicated in blue) are stored in a shift register. One column of the matrix $\boldsymbol{W'}_{MMSE}$ is transferred in each clock cycle. The subsequent intermediate matrices (indicated in green and orange) are added to the stored values. The sum of all three intermediate matrices is the final matrix product.

Downstream each of the matrix values is multiplied by the reciprocal of sigma which is derived from sigma beforehand.

For the calculation of $\boldsymbol{W'}_{MMSE}$ four submodules are used, which are assigned different columns of the resulting matrix. The input for each is the complete matrix $\boldsymbol{Q}_2$ and a row of $\boldsymbol{Q}_1$ (column of $\boldsymbol{Q}_1^H$). Internally, 12 vector products are calculated. This operation uses clock cycles as there are four columns per submatrix. The four submodules are fed with incoming data in a round-robin fashion. The result is collected by using valid signals. Finally, the matrix $\boldsymbol{W}_{MMSE}$ will be provided column wise to the following module, one column per clock cycle (see section 5.2.2.2).

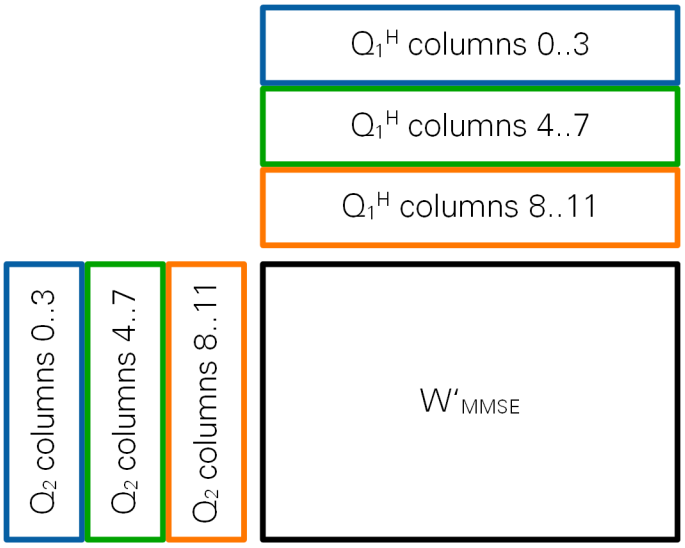

Figure 5-13: Matrices of Matrix Inversion Module

Note that the ZF equalizer matrix $\boldsymbol{W}_{ZF}$ is computed by the same modules as compared to $\boldsymbol{W}_{MMSE}$. The constant σ is chosen to be very small ($6.1 \cdot 10^{-5}$, smallest value which can be represented by the respective fixed point format) in this case.

5.2.3.6 MRC Weight Matrix Computation

The MRC equalizer is calculated according to (3.10) in parallel to the MMSE equalizer computation. The scaling factor $C_{m,m}$ defined in (3.13) can be derived from each column of $\boldsymbol{H}$ individually. The implementation reuses the Vector Normalization module N which has been introduced as part of the QR decomposition. Note that the Vector Normalization module normalizes with $\sqrt{(C_{m,m})}$. Hence, the normalization factors are applied twice sequentially. The last step is to compute the Hermitian matrix by calculating the conjugate complex of each vector element. The corresponding transpose operation is handled in the Channel Equalizer.

The MRC weight matrix computation is based on scaling all channel columns in a first step. However, the weight matrix needs to be provided to following modules row-by-row. Thus, the two already scaled submatrices of the MIMO channel matrix per RB are delayed in time to align them with the third submatrix. This allows to provide a row vector of 12 elements per clock cycle to the following module. This matches the MMSE equalizer matrix calculation output pattern.

The outputs of the MMSE equalizer computation and the MRC equalizer computation are time-aligned by internal delay modules to ensure that the results are available simultaneously.

5.2.3.7 Channel Equalizer

The channel equalizer module contains a memory to store the weight matrices corresponding to for all RBs, layers and antennas. This memory size is a constant regardless of the MIMO processor configuration. Increasing the number of antennas by

a certain factor results in a reduction of the number of RBs processed by a MIMO processor by the same factor such that the product of number of antennas and number of RBs is constant. The number of layers is fixed to 12, the maximum supported number. The number of used layers is varied by setting certain channel coefficients to zero as explained earlier.

The equalizer starts its operation as soon as I/Q data symbols are available. The respective weight matrix column is provided along with the data to four parallel processing units. Each of them calculates one vector $\hat{\boldsymbol{x}}$ (3.9) as there are four columns per submatrix. Each processing unit contains 12 dot products which calculate one element of $\hat{\boldsymbol{x}}$ in (3.9). After all columns of $\boldsymbol{W}$ are processed, four vectors of $\hat{\boldsymbol{x}}$, corresponding to four subcarriers, are available. Per clock cycle, the module provides four I/Q symbols, one from each vector, corresponding to the same spatial layer, at its output. That is, I/Q symbols corresponding to four subcarriers are forwarded layer-by-layer to the next module.

5.2.3.8 Scaling Factor Calculation

The weight matrix is also provided to the TX chain for precoding. Therefore, it is stored in a weight matrix memory as shown in Figure 5-8. Each value which gets written to the weight matrix memory is also fed into the scaling factor calculation block which is only used for precoding in transmit direction.

The scaling factor calculation computes one scaling factor per precoding matrix. The scaling factor normalizes the maximum transmit power of any transmit antenna in the digital domain to a value of 1. The scaling factor can be further adjusted by multiplication with an adjustable but static factor. This allows to increase or decrease the maximum transmit power per antenna, that is, to adjust a certain back off which may depend on the PAPR of the precoded signal.

The mathematical description can be found in (3.24). At first, the squared norm of each column of $\boldsymbol{W}$ is derived. Here, only the active layers are considered since the weights for all other layers are not valid (as they have been estimated in noise). The maximum squared column norm is stored in a register. After the last column was processed, the maximum value column norm value is used to calculate the reciprocal square root.

The resulting scaling factors are stored in the scaling factor memory.

5.2.3.9 Weight Matrix Memory and Scaling Factor Memory

The weights used for equalization and later for precoding as well as the scaling factors, used to scale the weight matrices are stored in memory. The precoding module which is described later reads the weight matrices and scaling factors form this memory.

There are three identical memory pages which can store a complete copy of weight matrices and scaling factors for all RBs in an OFDM symbol, each. The transmit and receive chains can access all three memory pages, in general. However, it is ensured that the transmit chain operates on a different memory page as compared to the receive chain at each point in time (mutually exclusive memory access), as follows:

- The transmit chain reads from the memory page which contains the latest complete copy of weight matrices, while the receive chain updates one the two other memory pages.
- The receive chain updates one of the two memory pages which is not accessed by the transmit chain.
- As soon as the receive chain has filled one memory page, it starts filling the other memory page.
- The transmitter requests access to the latest completely updated memory page. This transmitter request is coupled to transmitting a pilot OFDM symbol.
    - It remains on its current memory page if there is no update available.
    - It switches to one of the memory pages previously accessed by the receive chain if a new complete update is available.
    - The receive chain now starts updating the two memory pages which are not accessed by the transmit chain.

A common controller provides the memory page indices to be used to the transmit and receive chain.

5.2.3.10 Data Sink

As explained in section 5.2.3.7 the MIMO processor receive path provides four subcarriers per layer in three submatrices. The fixed-point format is 2.14 to be able to represent a M-QAM constellation. Those four subcarriers are written to a local FIFO. A serializer block takes those four values and divides them into two pairs of values to fit their binary representation into the U64 data format. In this format, the data are put into the P2P link which connects to the bit processor (see Figure 4-6). The data rate in I/Q symbols per second after the equalizer is derived in (5.4),

$$12\,\frac{I}{Qsymbols}\cdot\ 12\ layers\cdot\ 140\ \frac{OFDM\ symbols}{radio\ frame}\cdot 100\,\frac{PRBs}{OFDM\ symbol}\cdot\ 100\ \frac{radio\ frames}{s}\ =\ 201.6\cdot 10^{6}\,\frac{I/Q\ symbols}{s} \tag{5.4}$$

From (5.4), pairs of I/Q symbols can be transmitted at a rate of 100.8 MHz. This rate allows to use the default P2P writer clock rate (105 MHz, see section 4.3.4).

5.2.4 Transmit Path

Figure 5-14 shows an overview over the MIMO processor transmit path. The MIMO processor receives modulated frequency domain I/Q symbols from the bit processor at its input and delivers precoded frequency domain I/Q symbols to the RRHs splitters at its output.

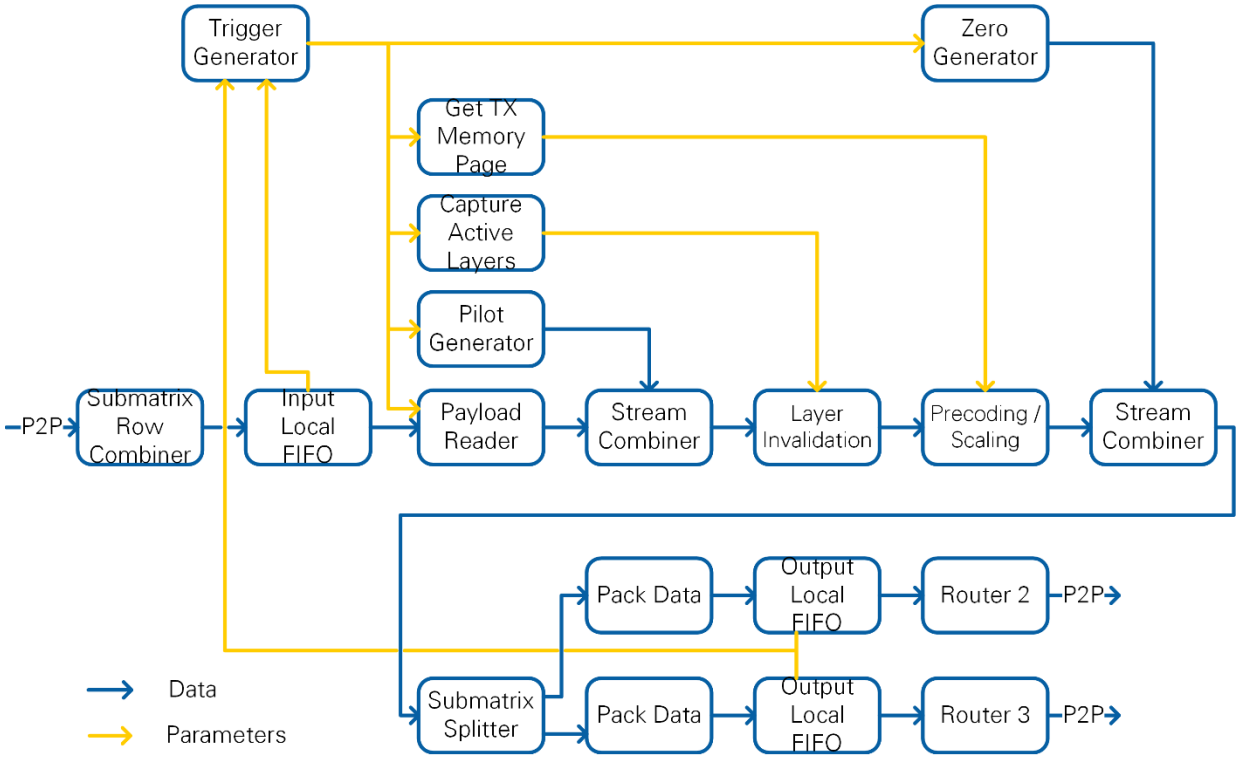

Figure 5-14: MIMO Processor Transmit Path

The MIMO processor operates RB-wise. That is, it starts processing if sufficient space is available in its output FIFOs and data corresponding to all layers and one RB is available in its input FIFO.

The MIMO processor multiplexes different data sources (zero generator, payload, pilots). The latency between the trigger generator output to the different data source modules and the submatrix splitter shown in Figure 5-14 is designed to be identical for all data sources to simplify the multiplexing operation.

Note that only one of the three data sources can be active at a time. The stream combiner selects and forwards the valid stream only.

Besides starting their operation based on trigger input and valid signals, each module uses internal counters to determine the position within OFDM symbols, for example the RB index within an OFDM symbol.

5.2.4.1 P2P Interface to the Bit Processor

The data source is composed of the Submatrix Row Combiner block and the Local FIFO shown in Figure 5-14.

The bit processor transfers data in the packet format described in section 4.3.2. Two U64 words provided by the bit processor are reassembled into a row of a submatrix (corresponding to four subcarriers).

5.2.4.2 Trigger Generator

The trigger generator provides trigger signals per RB to data source modules to start generating data worth one RB.

The amount of valid I/Q symbols increases as a matter of the precoding operation. The input to the MIMO processor is static and comprises 12 x 12 I/Q symbols per RB (matrix (4) in Figure 5-7). The output comprises *Max Number of Antennas* x 12 I/Q symbols per RB ((matrix (2) in Figure 5-7)). Thus, data source modules are not allowed to provide I/Q symbols in each clock cycle. The trigger generator triggers data source modules such that data is provided sufficiently slow to grant downstream modules enough clock cycles to complete their operation. This mechanism omits the need for handshaking within the transmit signal generation chain.

The trigger generator validates the fill state of input and output local FIFOs shown in Figure 5-14 as follows:

- Input local FIFO: In case a data OFDM symbol is to be generated for transmission, the input local FIFO needs to contain sufficient symbols to generate one RB for all layers (statically 12 x 12 = 144 symbols). This condition is not validated for all other OFDM symbol types.
- Output local FIFO: it is validated if enough space to capture one RB after precoding is available (matrix (2) in Figure 5-7). The processing delay of modules in-between the trigger generator and the output local FIFO is accounted for during the validation of available FIFO space.

Initially, the first trigger is generated if either the input local FIFO is completely full or the host manually enables the trigger generation.

The trigger generation module compares a local OFDM symbol index counter to the radio frame schedule to trigger one of the three data sources. There are three different trigger types, one for each data source. In addition, a trigger is generated to signal the OFDM symbol start.

Get Tx Memory Page block

Upon transmitting a pilot symbol, a link to the memory page containing the latest complete set of weight matrices and scaling factors is requested by this block. See section 5.2.3.9.

No updated link to the memory page is requested if the next OFDM symbol is a pilot OFDM symbol and the previous OFDM symbol was a pilot OFDM symbol as well. This restriction is imposed as these consecutive pilot OFDM symbols are used for tracking purposes at the receiver. Hence, the same precoding weights must be applied.

5.2.4.3 Data Sources

Upon receiving a trigger signal, all data sources provide I/Q symbols for one RB.

The pilot and data generation blocks provide I/Q symbols for all spatial layers per RB. The zero generator block provides I/Q symbols with value zero for all antennas per RB.

The I/Q symbols are provided row-by-row and submatrix by submatrix to downstream modules. Each data source ensures that idle cycles are added after providing each submatrix to downstream modules. These idle cycles ensure that downstream modules get enough processing time to finish their computation.

Pilot Generator
Generates frequency orthogonal pilot sequences for all spatial layers. Frequency orthogonality results in matrix (4) containing non-zero values only on its main diagonal.

Payload Reader
Reads payload data from the local input FIFO.

Zero Generation
Generates zero valued I/Q symbols for receive OFDM symbols, Sync OFDM symbols, and Guard OFDM symbols.

5.2.4.4 Layer Invalidation

The layer invalidation block overwrites I/Q symbols corresponding to inactive layers with zero values.

This block is controlled by the Capture Active Layers block. The host provides information which layers are active. Separate information is provided for pilot symbols and for data symbols. This information is read and updated upon the start of each pilot OFDM symbol. Depending on whether the next OFDM symbol is a pilot OFDM symbol or a data OFDM symbol, information about the active pilot layers or the active data layers is provided in terms of a length-12 Boolean array at its output. A TRUE value corresponds to an active layer.

5.2.4.5 TX Precoding

The Tx precoding block receives the index to the most recent weight matrix memory page and applies the weights contained therein.

Precoding is only started once a complete submatrix of matrix (4) has been received.

The processing operates in parallel on four subcarriers. Computing the signal to be transmitted from one antenna at one subcarrier requires to compute a dot product of length 12. A block, denoted Dot Product Group implements this operation in parallel for the four subcarriers. The Dot Product Group block can accept a new submatrix and a new row of the precoding matrix every 12 clock cycles. To guarantee a continuous output,

Twelve of these Dot Product Group blocks are instantiated in parallel. Note that these blocks are started one after the other. Each Dot Product Group block receives a new input (submatrix and row of the precoding matrix) every 12 cycles.

The precoding block provides matrix (2) at its output. One submatrix after the other and within each submatrix one row after the other. Each row corresponds to an antenna.

Therefore, a multiplexer cycles through the outputs of all 12 Dot Product Group blocks and routes one row from one Dot Product Group block to the output at a time.

Note that the precoding block operates at full precision. No truncation or rounding is performed.

5.2.4.6 TX Scaling

The Tx scaling block receives the index to the most recent scaling factor memory page and applies the factors contained therein. There is one scaling factor per RB.

Scaling is implemented as a multiplication, followed by rounding to the complex fixed point 2.14 format. Any overflows are counted for evaluation on the host.

5.2.4.7 Data Sink

The data sink comprises all modules between submatrix splitter block and P2P FIFOs, as shown in Figure 5-14.

The first operation is carried out by the Submatrix Splitter block. It splits the incoming data stream into two parallel streams. The first stream corresponds to the first half of transmit antennas and the second stream to the second half of transmit antennas. Note that at this point signals for all antennas the MIMO processor is designed for are generated, even though the actual number of transmit antennas may be less.

To implement this operation, the complete submatrix is first stored in a local memory. Both streams are fed alternating as follows. Two rows comprising eight I/Q symbols (four subcarriers, two antennas) are forwarded to stream 1, followed by forwarding two rows to stream two and so forth. Each forwarding operation takes two cycles but leaves four cycles for processing for the respective stream (as the other two cycles are used to feed the other stream).

The packets of eight subcarriers are packed into three U64 words as explained in section 4.3.1. This operation takes three cycles.

Routers 3 and 4 distribute packets of three U64 words to different RRH subsystems, depending on which packet corresponds to which antenna. This mapping is defined by a router program. See section 5.1.6.

5.3 Bit Processor

5.3.1 Overview

The bit processor acts as interface between host data source and sink and MIMO processors. It connects to up to four MIMO processors. In the transmit path it accepts binary payload from up to 12 data sources from the host and provides respective modulated symbols to the MIMO processor. The receive path operates vice versa.

5.3.2 FPGA Considerations

5.3.2.1 Routers

The bit processor uses the same U64 routers as described in 5.1.6 for the data exchange with MIMO processors. Data interface format and data rates correspond to the respective description in section 5.2.3.10.

5.3.2.2 Data Processing

The I/Q symbol rate as calculated in (5.4) can be handled using a clock rate of 205 MHz without need for parallel computations. This clock rate is slightly higher than needed according to (5.4). The payload part, therefore, must handle one byte per clock cycle as 256-QAM 8-bits are mapped to one I/Q symbol. The additional idle clock cycles are used in part, for example, for control overhead in state machines or for processing header information. The difference between clock rate (205 MHz) and I/Q symbol rate (201.6 MHz) leaves 20 clock cycles per spatial layer and OFDM for overhead operations as calculated in the following equation.

$$\frac{205\ \text{MHz} - 201.6\ \text{MHz}}{12\ \text{layers} \cdot 140 \frac{\text{OFDM symbols}}{\text{radio frame}} \cdot 100 \frac{\text{radio frames}}{\text{s}}} = 20.2 \tag{5.5}$$

5.3.3 Receive Path

The building blocks of the receive path are shown in Figure 5-15.

As for the MIMO processor, the receive path does not implement back-pressuring. The initial throttle module guarantees enough clock cycles for downstream modules. The limiting module is the Packet Validator (consumes the most overhead clock cycles).

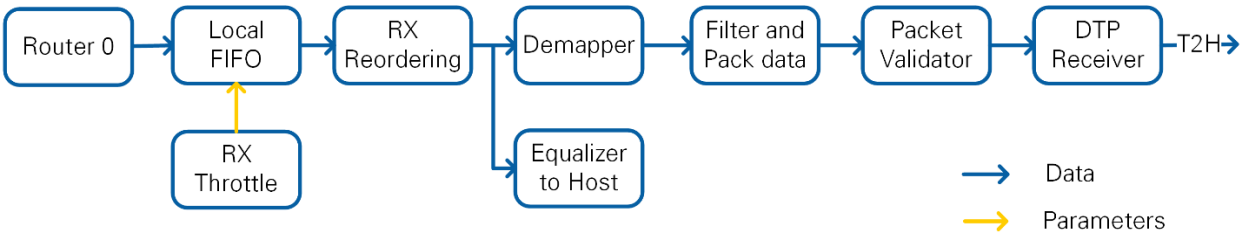

Figure 5-15: Bit Processor Receive Path

5.3.3.1 Router

The router functionality is described in 5.1.6. The program depends on the actual number of MIMO processors. Each RB is read from one MIMO processor as described in section 4.3.2. The output is stored into a local FIFO. The FIFOs dimension is chosen to hold at least one RB of data (see RX throttle module in next subsection).

5.3.3.2 RX Throttle

This module checks the local FIFO for availability of one RB of data. Once this condition is fulfilled, the RB is read from the FIFO. The Rx reordering block takes 144 cycles per RB to complete its operation. Accordingly, the Rx Throttle block provides a new RB to the Rx

reordering block 144 cycles after the start of the previous block, at earliest. After finishing reading one OFDM symbol, a waiting time of 120 clock cycles is applied before the first RB of the next OFDM symbol is read. This waiting time is important for overhead of the Packet Validator block.

5.3.3.3 RX Reordering

The Rx Reordering block receives RBs which contain 12 interleaved layers according to the format in Figure 4-6. One hundred of these RBs are read into a two-page memory. While one page is filled, the other page is read. An internal counter computes when the write operation is completed. This event serves as a trigger for the read operation.
Upon having written data for one OFDM symbol and 12 layers to a memory page, the reordering operation starts. The module reads data in such an order from the memory that complete OFDM symbols, corresponding to a single layer, are forwarded to downstream modules. That is one OFDM symbol corresponding to layer 1 is forwarded followed by one OFDM symbol for layer two, and so on.

5.3.3.4 Equalizer to Host

For plotting constellations at the host, the reordered I/Q samples of one OFDM symbol per radio frame are transferred to the host. The OFDM symbol index can be set from the host. The module ensures that there are enough free elements in the T2H FIFO to not to lose data. The target-to-host (T2H) FIFO is designed to hold a complete OFDM symbol. The host can therefore assume a fixed length array per radio frame.

5.3.3.5 Demapper

The incoming I/Q samples are mapped to LLR values and subsequently to hard bits, assuming 256-QAM modulation. No knowledge about the frame schedule is needed for this operation. The bits are Gray coded (compare to the IEEE 802.11 a standard). The output format is U8 which contains the 8 bits represented by 256-QAM. The valid bits, in case a different modulation scheme was used, are extracted in following modules.

Internally, the LLR demapper compares the I/Q samples to the decision boundaries. It returns the result as softbits, which represent a probability of the bit being 0 or 1. The sign of this bit is used to decide for 0 or 1.

5.3.3.6 Packet Separator

The packet separator is an internal module which is used at multiple places within the bit processor. It reads the frame schedule memory and generates information about transport blocks in the frame schedule. That is, per 8 bits (that is, per I/Q symbol) it generates information about Layer ID, OFDM symbol index and modulation format. This information is provided in an output FIFO for other modules.

The packet separator block implements filter mechanisms. In the downlink, OFDM symbols corresponding to the uplink are skipped and vice versa for the uplink. It is also possible to only keep just one layer or just layers with non-empty modulation set.

The module starts processing as soon as the frame schedule memory is written from the host.

5.3.3.7 Filter and Pack Data

This block uses a packet separator block to align the data stream with frame schedule-related information. It is required since the incoming byte stream from the demapper is only available for data symbols. The bytes are split into individual bits. The modulation of the current transport block is used to keep only the bits which correspond to the modulation scheme used in this OFDM symbol. These bits are grouped into bytes in a second step. Note that each transport block aligns with a byte boundary for the payload.

Layers where the modulation is set to *none* are discarded by this module by not forwarding the valid signal.

5.3.3.8 Packet Validator

The packet validator contains multiple submodules. It terminates transport blocks for unused layers, reverts scrambling, checks the transport block CRC and filters the transport blocks based on the CRC result.

In a first step, a packet separator is used to align transport block related information with the data stream.

The **RX Layer Termination** block discards bytes corresponding to layers which are not active on this system. This functionality is used to deactivate layers on the MIMO MS (otherwise other layers than the own ones can be received as well).

The **Packet Scrambler** applies de-scrambling based on OFDM symbol index, layer and modulation.

The **Packet Validator** calculates the CRC32 checksum over a transport block and verifies the remainder after last byte of the transport block. Additionally, the transport block header is decoded. The indicated payload length is checked against the maximum payload size of the current transport block. At the end of each transport block a cluster with the check results and the header contents is provided to downstream modules.

The **Packet Filter** receives all bytes of a transport blocks and the corresponding packet information from the packet validator which it stores in two separate FIFOs.

Packets with invalid CRC result or incorrect header are read from the FIFO but not forwarded to the output.

The packet information from the packet validator is also forwarded to the output. There is an additional valid flag indicating which bytes belong to the payload of the packet. This is required for the DTP decoders.

5.3.3.9 DTP Receiver

The DTP receiver detects DTP packets within the incoming byte stream. Twelve DTP receiver submodules operate independently and in parallel. Each DTP receiver submodule

corresponds to a data source. The output of the 12 DTP receivers is transferred in time multiplex through a single FIFO to the host.

The layer ID is extracted from the packet information which accompanies a transport block at the input to the DTP receiver. Layer-to-data source mapping information is provided to assign the layer to the data-source-specific DTP receiver submodule.

The DTP receiver subblock checks the validity of the DTP header and the DTP Sync End field (see Table 2-2 for the DTP packet structure). It forwards the payload part of the DTP packet along with information whether the packet is to be considered valid.

Each DTP receiver subblock is followed by a DTP FIFO collect block. This block forwards only valid DTP packets (CRC check ok, Sync End word found). A token circulates between the 12 DTP FIFO collect blocks. The block which has received the token first generates and forwards an RX indication header and then forwards the payload of a correct DTP packet for transmission to the host. After completing this operation, or if no DTP packet is available, it passes the token to the next DTP FIFO collect block. This mechanism ensures that only a single DTP FIFO collect block forwards valid DTP packet payload at a time.

The output of the 12 DTP FIFO collect blocks is fed into an OR operation. The single output of this operation is fed into a T2H FIFO for transferring the receive indications to the host.

5.3.3.10 Packet Counters

Based on the packet information provided by the packet filter, the following metrics are derived per spatial layer:

- Number of valid MAC payload bytes
- Number of valid PHY transport block bytes
- Total number of transport blocks
- Number of transport blocks with CRC pass

These metrics are used at the host for throughput and BLER calculations.

5.3.4 Transmit Path

The bit processor transmit path accepts DTP packets from the host at its input and delivers modulated symbols for all 12 spatial layers at its output for further processing at the MIMO processor. The transmit path implements transport block generation, modulation and routing to up to four MIMO processors.

Figure 5-16 shows an overview over the bit processor block diagram. The dotted lines in Figure 5-16 separate three processing partitions, where each partition implements back pressuring (see following subsections for details).

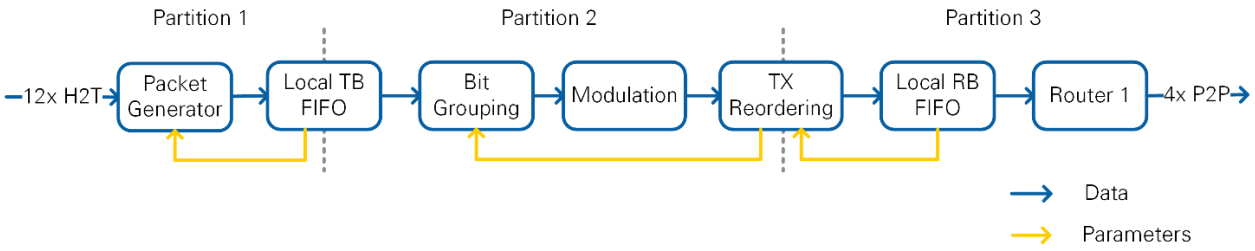

Figure 5-16: Bit Processor Transmit Path

5.3.4.1 Packet Generator

The packet generator block implements the functionality described in sections 2.3.2 - 2.3.4. It receives DTP packets from up to 12 data sources from the host. Each data source corresponds to a mobile station. Each DTP packet is mapped to layers which are assigned to a mobile station. At each layer, transport blocks are generated from (partial) DTP packets, based on the radio frame schedule.

A packet separator block (see section 5.3.3.6) parses the frame schedule to find transmit data OFDM symbols and spatial layers, for which transport blocks of a certain length need to be generated. A state machine triggers the generation of a transport block from DTP packets, if there is sufficient space available in the local transport block FIFO. This triggering must be enabled from the host.

Upon receiving the start trigger from the state machine, the packet content generator block starts generating one transport block of correct length. The packet content generator block has knowledge which data source (FIFO) corresponds to the layer for which it is generating a transport block through a mapping table provided from the host. It reads the number of available elements from the selected data source FIFO. Based on this information it determines the maximum payload size for the transport block and generates the transport block header. The next step is to forward the payload of determined length from the data source FIFO to the output of the module. Additionally, the packet content generator block counts the transmitted PHY (transport blocks) and the MAC (DTP packets) bytes per layer. These metrics are used at the host for throughput calculations.

Downstream, the FCS Add block attaches the CRC to the transport block. At this point, the transport block is complete. It is then scrambled based on modulation, OFDM symbol index and spatial layer index and stored in the local transport block FIFO.

5.3.4.2 Bit Grouping

The Bit grouping block applies the reverse operation of the Filter and Pack block in the receive path described in section 5.3.3.7. It splits the transport block bytes into individual bits and regroups them depending on the modulation scheme used for the transport block. Therefore, it uses a packet separator block to find all transport blocks for data symbols within the frame schedule and the corresponding modulation scheme.

The TX Bit grouping starts its operation upon receiving a ready for next OFDM symbol information from the TX reordering module. This ensures the back pressuring in the second partition of Figure 5-16.

After reception of the trigger a transport block for each of the 12 layers is processed as soon as the transport block payload is available in the local TB FIFO. The payload is read from the FIFO using a modulation specific pattern so that the downstream Modulation block is not overloaded. Each payload byte is split into separate bits. The bits are regrouped based on the modulation, for example, one payload byte results in four transfers of 2-bit values in case of QPSK. Zero valued transfers are executed for layers with modulation set to none (inactive layers). From this point on 12 spatial layers are processed statically for all transmit OFDM symbols, even though only a subset of spatial layers may be active.

Information about the modulation format is provided at the output of the block in parallel to the transport block data.

5.3.4.3 Modulation

Groups of bits are mapped to modulated I/Q symbols, depending on the modulation information provided by the TX Bit grouping block. In case the modulation format is set to *none*, zero valued I/Q symbols are generated.

5.3.4.4 Reordering

The TX reordering block changes the order of I/Q symbols from transport block to resource blocks. The input is provided layer by layer and I/Q symbol by I/Q symbol. The output pattern is described in section 4.3.2.

The TX reordering block contains two memory pages. The first page is used for writing while the second page is read from. When the reading and writing operation for one OFDM symbol is complete, the former read page becomes the new write page and vice versa for the former write page. This event also triggers the upstream module to generate a new OFDM symbol for transmission. Initially (on system start up), the first memory page serves as write page and is declared to be empty which triggers generation of the first OFDM symbol.

The read operation is done per RB. If the local RB FIFO provides sufficient space for an RB of data, the read operation of the RB starts.

5.3.4.5 Router

The router distributes the resource blocks to the MIMO processors as described in section 4.3.2. It uses the common router block described in section 5.1.6.

6 Single-Antenna Mobile Station FPGA Implementation

6.1 Overview

Figure 6-1 shows an overview of the MS architecture implemented on the USRP device. The architecture comprises the analog part and the digital baseband part, where the latter is in the scope of this section.

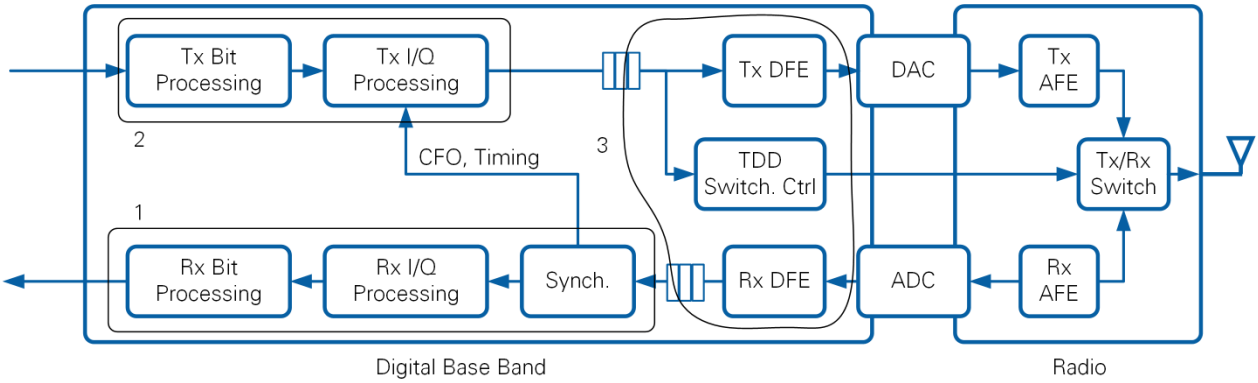

Figure 6-1. Single-Antenna MS FPGA Processing Overview

The digital baseband FPGA implementation is structured into three independent loops, indexed 1…3 in Figure 6-1.

1) The first loop covers the receive signal and bit processing and is further explained in section 6.2.
2) The second loop covers the transmit signal and bit processing and is further explained in section 6.3. The transmit processing loop receives synchronization information from the receive processing loop.
3) The third loop covers the digital frontend. It is connected to the first two loops through local FIFOs. It contains the control for TDD switching which is briefly covered in section 6.3. The digital frontend itself is standard USRP functionality and is not further covered.

The digital frontend is connected through ADC and DAC with the radio transceiver part which implements up-/down conversion, gain stages, for instance. The transmit/receive switch in the radio transceiver part is controlled form the digital baseband.

Note that the loop 1 and 2 use a clock rate of 192 MHz, that is, they are operate at about six times the symbol rate of 30.72 MHz.

6.2 Receive Path

Figure 6-2 shows the functionality implemented in the receive path. It comprises the following three main functional blocks:

- Synchronization

- IQ processing (time and frequency domain)
- Bit processing (RX bit processing and packet validation)

These three blocks are part of the *DL RX baseband & bit processing* FPGA top level loop. This section covers these main three functional blocks in detail.

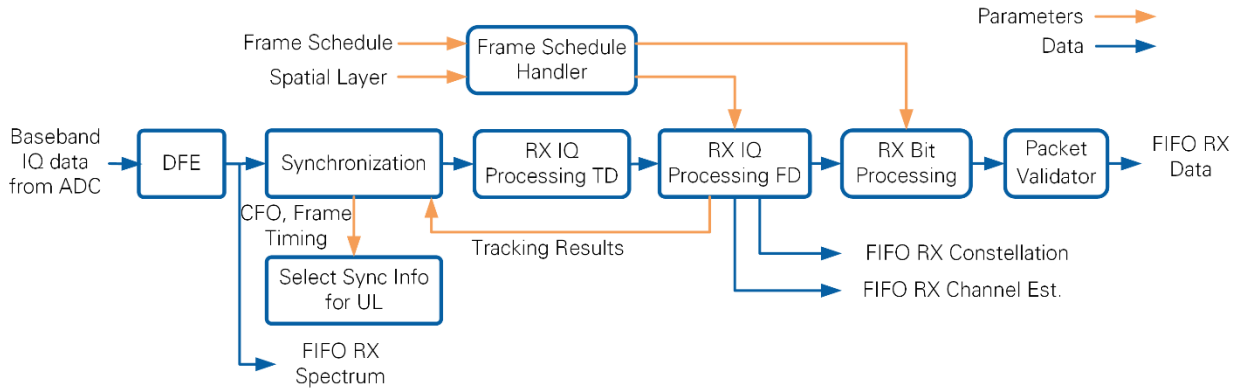

Figure 6-2. Single-antenna MS Receive Processing Implementation Overview

Note that the digital frontend contains standard USRP functionality and is not further covered here. More information can be found within the NI-USRP Simple Streaming sample project.

6.2.1 Synchronization and Tracking

This subsection explains implementation details of the *Synchronization* block depicted in Figure 6-2. Figure 6-3 presents the synchronization signal processing block diagram. The main purpose of these blocks is to compute carrier frequency offset correction and frame timing adjustment parameters once per radio frame and to apply them at the start of the next radio frame. The synchronization block provides time and frequency aligned time domain I/Q samples at its output. The respective synchronization algorithms have been introduced in section 3.7.

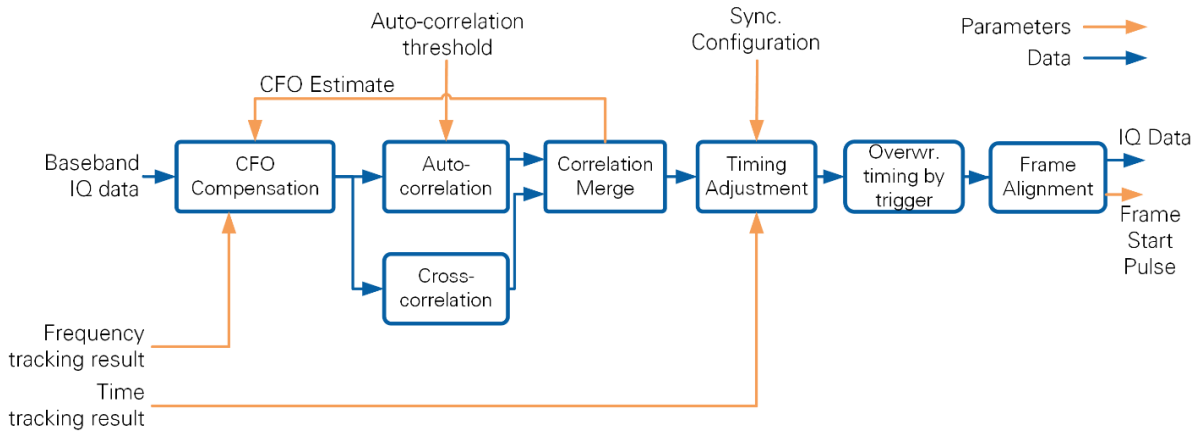

Figure 6-3. Synchronization Implementation Block Diagram

The carrier frequency offset is compensated in a first step in the **CFO Compensation** block. The respective parameters are computed in subsequent blocks and are fed back into the compensation block. The block operates in the following two modes:

- Before successful initial synchronization:
  The CFO is estimated in the **Autocorrelation block.** The CFO estimate is updated based on an estimate in the current frame. The updated estimate is applied from the start of the next radio frame. The updated estimate is obtained by adding the update to the CFO estimate from the previous radio frame. A configurable weight (*CFO Factor*) is applied to the update before adding it, except for the very first estimate where a weight of 1 is chosen.
- After successful initial synchronization:
  The residual CFO is estimated within the current frame in a tracking block described later. The residual CFO is accumulated (over multiple radio frames) and added to the initial estimate described before. This updated estimate is used to compensate the CFO in the next radio frame.

CFO compensation is followed by cross and autocorrelation blocks which compute initial time and frequency synchronization parameters.
The **Cross-Correlation** block detects the PSS contained once per radio frame. Cross-correlation is implemented by two finite impulse response (FIR) filters (real and imaginary parts). This operation is executed on a reduced sample rate of 1.92 MS/s, which is the result of a decimation by 16. The value 16 is obtained from 2,048/128, where 128 is the required IFFT-order to create the PSS signal. Peak detection and peak validation blocks verify the conditions listed in section 3.7.1 as part of the cross-correlation block

The **Autocorrelation Block** continuously computes the autocorrelation of the CP and the end of the respective OFDM symbol. The autocorrelation is computed at the full sample rate of 30.72 MHz. Its purpose is to locate the OFDM symbol boundaries. The autocorrelation value is calculated by multiplying the I/Q samples with delayed and conjugated I/Q samples followed by accumulation. A division by the energy normalizes the resulting value, denoted as timing metric, which ranges from 0 to 1. The timing metric magnitude is valid if it exceeds a specified threshold. This threshold is set to 0.5 by default but can be configured within the MS Host code.

The **Correlation Merge** block checks if the distance between the valid peaks found by cross-correlation and by autocorrelation block is smaller than ± 16 samples (corresponds to ± 1 sample at the reduced sampling rate used for cross-correlation). A radio frame is successfully detected if this condition (denoted *frame detection condition*) is fulfilled. The correlation peak found by autocorrelation determines the OFDM symbol start. The correlation peak found by cross-correlation determines the first OFDM symbol of a radio frame as explained in the description of the following block.

The **Adjust Timing** block uses the knowledge of the PSS position within a radio frame to compute the start of the first sample of a radio frame. The block operates in the following two modes:

- Before successful initial synchronization:
  The estimate of the first PSS sample delivered by the **Correlation Merge block** is subtracted from the known index of the PSS start index within a radio frame to determine the radio frame start. A configurable timing advance can be subtracted. That is, a positive timing advance parameter shifts the OFDM symbol start from the first sample of the OFDM symbol into the preceding cyclic prefix.
- After successful initial synchronization:
  It can be configured as part of the host code whether tracking results computed by the time tracking module are used. If this is the case, the frame timing estimate described before is kept static. It is updated by time tracking updates per radio frame. If time tracking is disabled, the frame timing relies on the initial frame timing estimate which is updated continuously per radio frame.

The **Overwrite Timing by Trigger** module provides a radio frame start pulse at its output which indicates the start of a radio frame. It uses a local counter which increases with every data element in comparison to the given frame start timing. In addition, the block allows to select among the following two options for frame timing synchronization:

1) Normal: The timing estimate previously described is used
2) Bypass: the start of the receive signal processing has been triggered (through a cable) from the base station. The radio frame start is known in this case (first received sample) and the frame timing estimate estimated as described before is not used.

Upon receiving a radio frame start pulse, the **Radio Frame Alignment** masks the next 307,200 incoming samples as valid (radio frame length). Note that in case the radio frame was not correctly detected, the incoming samples would be masked as invalid by this module.

Note that the timing and frequency adjustment information is passed to the uplink transmitter for adapting the uplink transmit frame timing and frequency correction.

Uplink transmit synchronization through the downlink: Furthermore, the two uplink transmitters of a USRP device derive their frame timing and frequency correction from the downlink synchronization. The downlink synchronization blocks of each receive chain pass their results into the **Select Sync Info for UL block** shown in Figure 6-2**.** It can be selected whether the downlink synchronization results corresponding to the first or second USRP device antenna are used for the uplink. The **Select Sync Info for UL block** forwards the selected sync result to the uplink transmitter.

6.2.2 Time and Frequency Domain I/Q Processing

Time and frequency domain I/Q processing corresponds to *RX IQ Processing TD* and *RX IQ Processing FD* blocks in Figure 6-2. These two blocks take time and frequency synchronized radio frames at their input and deliver equalized QAM symbols at their output. Figure 6-4 shows the individual subblocks.

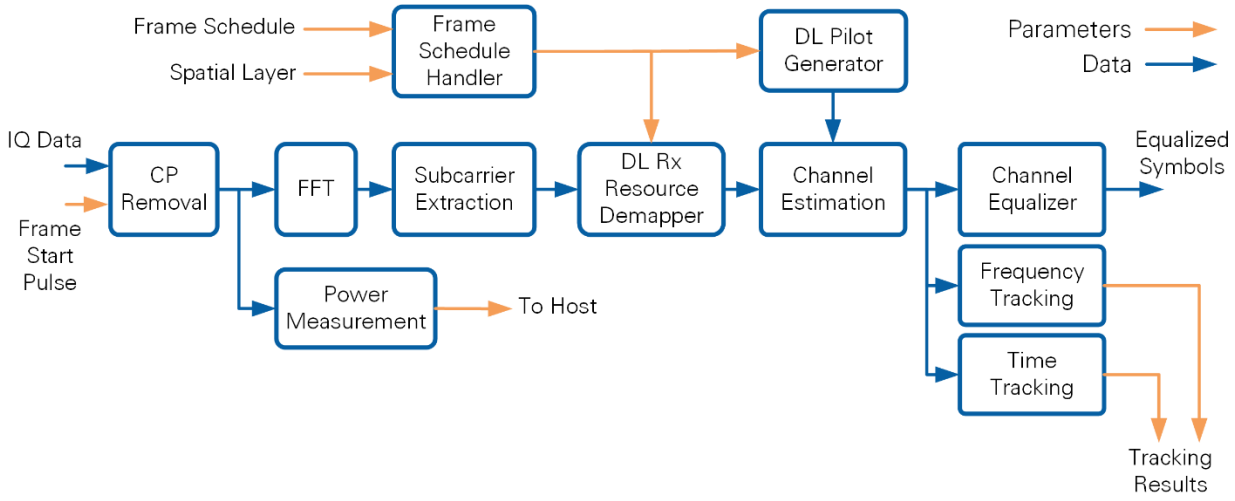

Figure 6-4. Receive I/Q Processing Block Diagram

The **CP Removal block** removes the valid flag from all CP samples in a radio frame.

The **FFT block** converts 2,048 time domain samples to frequency domain using a Xilinx FFT. A continuous phase shift between samples by 180 degree (toggling negation) corresponds to a cyclic FFT shift and shifts the DC carrier to the 1,024th output sample of the FFT, that is, the middle carrier corresponds to the DC carrier.

The **Power Measurement block** computes the power per OFDM symbol. It provides the maximum power detected in any OFDM symbol of a radio frame at its output. Also, it provides the power of a selected OFDM symbol at its output. These values are used at the host to compute AGC and uplink power control adjustment values.

The **Subcarrier Extraction block** removes the valid flag from all, except for the inner (occupied) 1,200 subcarriers out of the 2,048 subcarriers. Also, the DC subcarrier is masked as invalid.

The **Frame Schedule Handler block** uses a priori knowledge about the frame schedule and the spatial layer to assign an OFDM symbol type, such as *downlink data* or *downlink pilot* to each OFDM symbol within a radio frame. Also, the modulation type, such as *16-QAM*, is assigned to each data OFDM symbol for later use.

The **DL Rx Resource Demapper block** uses the OFDM symbol type to assign a symbol type, such as *DL data, ZERO,* or *DL pilot*, to each subcarrier in an OFDM symbol.

The **DL Pilot Generator block** provides transmitted frequency domain pilot signals corresponding to the layer which is used for transmission at this mobile station.

These pilot symbols are used in **Channel Estimation block**, which implements the channel estimation functionality described in section 3.4.1. Note that channel estimation is implemented in two steps. First, the channel is estimated at pilot subcarrier positions. Second, the channel at all occupied subcarriers is computed through linear interpolation.

The **Channel Equalizer block** uses the channel estimate provided by the **Channel Estimation block** to equalize the channel per subcarrier as described in section 3.4.1.

The **Time Tracking block** uses the phase of the first channel estimate in a radio frame to determine a residual frame timing error as described in section 3.7.2. It coerces the error to ± 1 sample for $|\text{error}| > 1$ sample and 0 else. The coercion determines the step size of the time tracking algorithm. This coerced residual error is fed back to the **Adjust Timing block** described in the section 6.2.1.

The **Frequency Tracking block** uses the phase difference of the first two channel estimates in a radio frame to determine a residual carrier frequency offset as described in section 3.7.2. It coerces the error to ± 10 Hz for $|\text{error}| > 10$ Hz and 0 else. The coercion determines the step size of the frequency tracking algorithm. This coerced residual error is fed back to the **CFO correction block** described in the section 6.2.1.

The equalized data symbols are forwarded to the Rx Bit Processing block described in the next section.

6.2.3 Bit Processing

Figure 6-5 illustrates the different bit processing steps. Bit processing accepts equalized symbols at its input and provides all errorless decoded MAC packets at its output.

Note that the bit processing is like the base station operation of the bit processor.

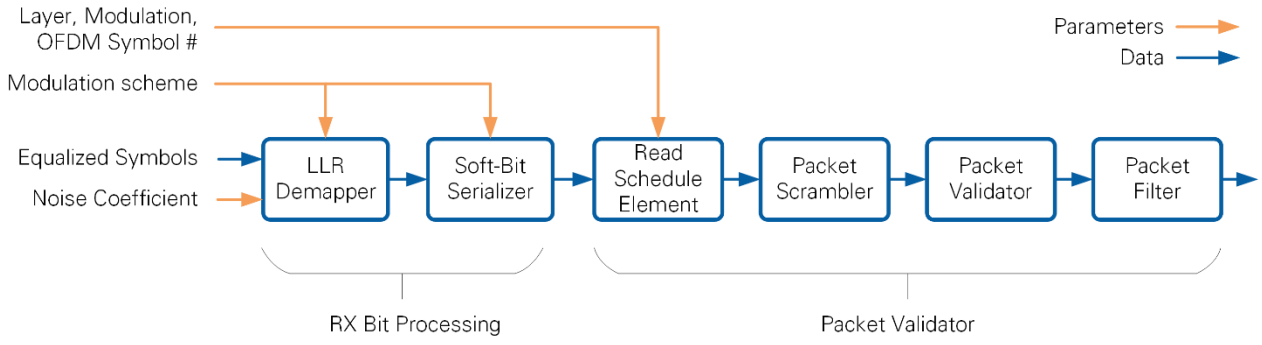

Figure 6-5. Single-Antenna MS Bit Processing

The different processing steps are as follows.

The **LLR Demapper block** computes an array of log likelihood ratios (soft bits) from each equalized symbol, identical to the LLR demapper described in section 5.3.3.5. The LLR

demapper is provided static information about the noise variance, which is configured to be 40, corresponding to a SNR of about 16 dB.

The **Soft Bit Serializer block** consumes an array of soft bit values per equalized symbol and delivers them sequentially to the following block. The Soft Bit Serializer block is followed by a hard bit decision based on the sign of the soft bit and the grouping of 8 bits into one byte.

The **Read Schedule Element block** aligns bytes belonging to one OFDM symbol with respective control information (layer, modulation, OFDM symbol number).

Packet **Scrambler, Validator** and **Filter blocks** operate similarly, introduced in section 5.3.3.8. The output is a receive indication which is evaluated by the next block.

Two single antenna mobile stations are implemented on one USRP device. Both implement the bit processing shown in Figure 6-5 individually. The output of the packet filter block is fed into the **DTP Receiver block** (not shown in Figure 6-5) for both mobile stations jointly. For each mobile station, the payload is extracted and marked as valid depending on finding correct synchronization words at predefined positions. Payload packets (embedding the information to which user they belong) corresponding to two users which are marked as valid are interleaved on a packet by packet basis and sent to the host.

This operation is similar to that of the BS (see section 5.3.3.9), except for that two MSs are mapped to one DTP submodule block.

6.3 Transmit Path

Figure 6-6 shows the functionality implemented in the transmit path. The transmit path accepts binary data at its input and delivers a time domain waveform at DAC sampling frequency at its output. The transmit processing comprises three main functional blocks

- Bit processing (TX bit processing and packet validation)
- I/Q processing (time and frequency domain)
- TDD switching control

These main three functional blocks are further described in this section.

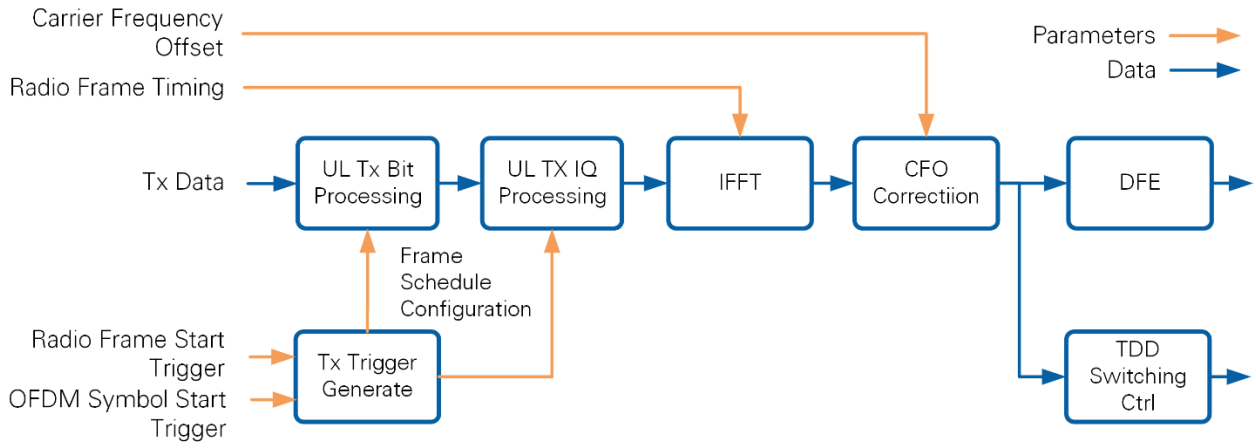

Figure 6-6. Single-Antenna MS Transmit Processing Implementation Overview

Note that the digital frontend contains standard USRP functionality and is not further covered here, similar to the receive digital frontend. More information can be found within the NI-USRP streaming example project.

The start of a radio frame is derived in the MS downlink receiver. This frame start is provided the MS uplink transmitter. A start trigger for the processing of each OFDM symbol is derived from the radio frame start trigger. Both trigger signals are passed to the **Tx Trigger Generation** block. This block has knowledge of the radio frame schedule. It provides a start trigger per OFDM symbol for bit and I/Q processing. Note that the trigger for the I/Q processing is sufficiently delayed as compared to the trigger for the bit processing such that it is guaranteed that the bit processing for one OFDM symbol is finalized before the I/Q processing starts. Also, it provides frame schedule configuration information, such as the assigned spatial layer, OFDM symbol number and type, and modulation scheme. Note that this frame schedule information triggers processing for uplink and for downlink transmission. That is, uplink transmit bit processing and I/Q processing remain active if a certain OFDM symbol is allocated to the downlink. A zero-valued signal is generated in this case.

6.3.1 Bit Processing

Figure 6-7 shows a block diagram of the transmit bit processing. The bit processing is like the base station bit processing presented in section 5.3.4.

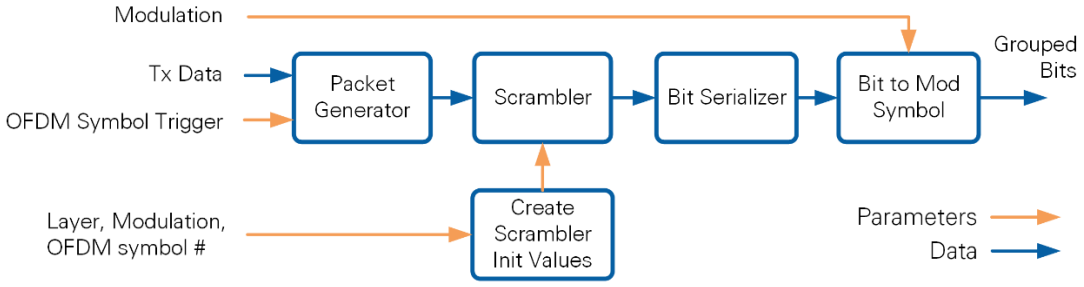

Figure 6-7. Single-Antenna MS Transmit Bit Processing

Upon receiving and OFDM symbol trigger, the **Packet Generator block** reads binary data from the host and formats it into a MAC packet according to the format described in section 2.3. That is, a header is prepended and padding and a frame check sequence are appended. Note that this transport block is mapped to a single OFDM symbol in the following modules.

The **Scrambler** block scrambles the binary data. It receives initialization information per OFDM symbol. The initialization information is derived from the spatial layer number, the OFDM symbol number and the modulation scheme. The scrambling polynomial is chosen according to 3GPP LTE specifications [3].

The **Bit Serializer** block receives scrambled bytes at its input and forwards individual bits at its output. The **Bit to Mod Symbol** block groups sufficient bits per modulation symbol and forwards them to the I/Q processing block for mapping to modulation symbols.

6.3.2 I/Q Processing

Figure 6-8 shows the transmit I/Q signal processing block diagram. The processing chain accepts information from the TX Trigger Generation block and grouped data bits at its input and delivers a time domain wave form at 30.72 MHz symbol rate at its output.

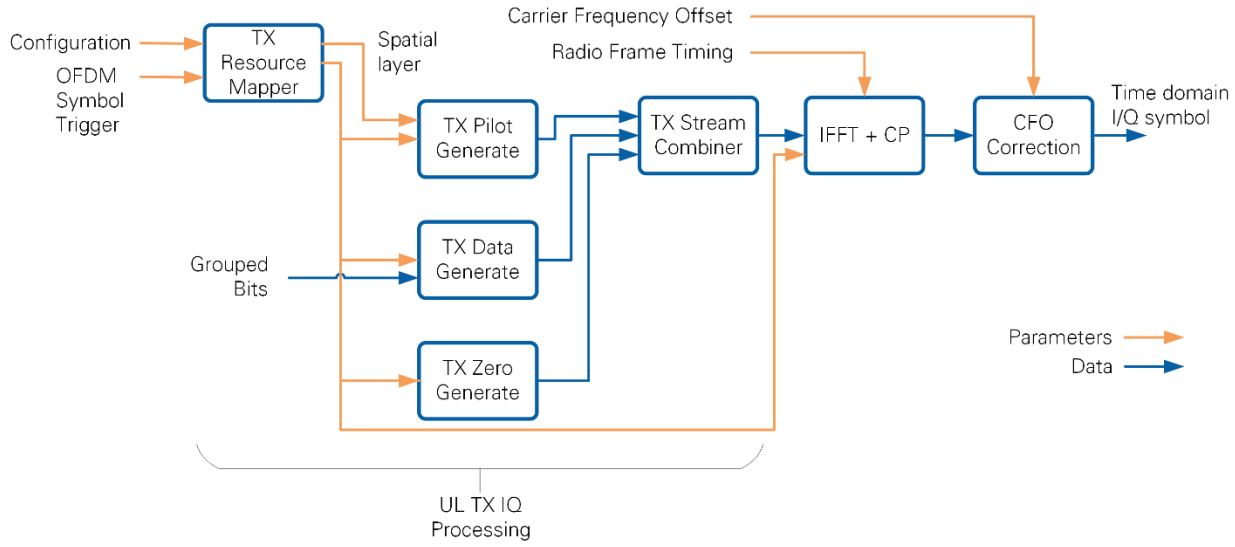

Figure 6-8. Single-Antenna MS Transmit I/Q Processing

The **TX Resource Mapper block** is fed with information about the spatial layer ID and the OFDM symbol type from the TX trigger generation block. It generates control signals for each downstream data source and the IFFT.

The **TX Pilot Generate block** generates QPSK modulated pilot symbols. Information about the spatial layer is used to index a predefined random pilot sequence. Note that every twelfth subcarrier is occupied by a pilot. The remaining subcarriers within a pilot OFDM symbol are filled with zeros by the **TX Zero Generate block**.

Note that the three signal generation blocks generate signals for 1,200 used subcarriers. The remaining subcarriers are filled with zeros through a separate valid signal for the IFFT input from the TX resource mapper block. Note also, that at most one of the three blocks is activated for a certain subcarrier. Inactive blocks provide zero on their output.

In case that OFDM symbol type corresponds to the downlink (receive symbol for the mobile station), zeros are generated. That is, the transmit I/Q processing generates a signal also for receive OFDM symbols. This signal contains only zero values.

The **TX Stream Combiner block** applies an OR operation to the three input signals to generate a single output signal.

The **IFFT + CP block** computes the time domain wave form using a length-2,048 XILINX IFFT. In addition, a CP is prepended. The CP length is varied according to the radio frame timing information. This timing information is provided by the mobile station synchronization receive algorithm as discussed in section 6.2. The respective timing correction information is derived with the radio frame start trigger. Note that the timing correction is applied once per radio frame during the first CP in a radio frame. The IFFT operation is followed by inverting the sign of every second time domain symbol. This operation implements an FFT shift.

The **CFO Correction block** receives the CFO measured in the MS downlink receiver. This CFO is corrected before the uplink transmission such that multiple mobile stations are received frequency-synchronous at the base station.

The CFO-corrected time domain symbols are forwarded to the digital frontend FPGA loop, which, for example, implements resampling to the DAC sampling frequency.

6.3.3 TDD Switching

TDD switching control is implemented as part of the DFE loop shown in Figure 6-1. The implementation is identical to the BS. Please refer to section 5.1.7.

6.3.4 Receive-Transmit Timing Calibration

The transmitter and receiver processing are time-calibrated such that the uplink and downlink radio frame start are identical at the physical antenna port of the base station. In general, the uplink transmit timing is derived from the downlink receive synchronization. Calibration values are used to determine the uplink transmit timing from the receive timing. These calibration values account for processing delays within the signal processing chain. This principle is illustrated in Figure 6-9.

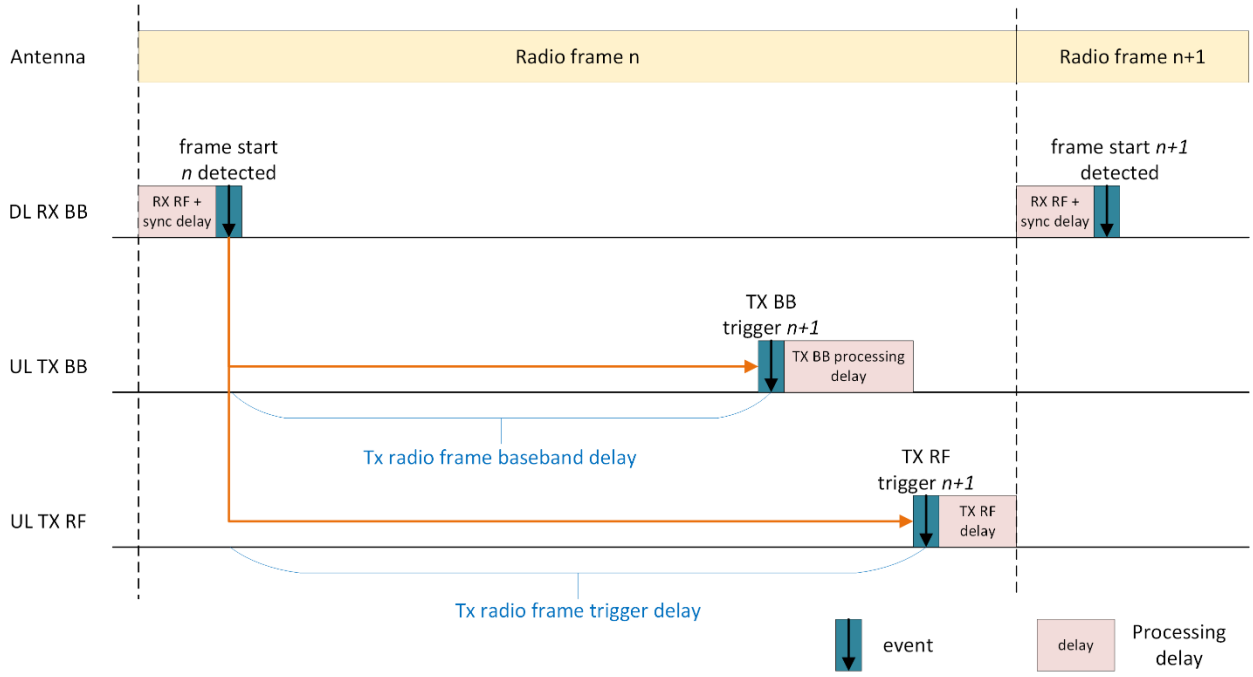

Figure 6-9. Receive-Transmit Timing Calibration at the Single Antenna Mobile Station

The transmit timing is determined in the following steps:

- The synchronization algorithm in the DL baseband receiver estimates the radio frame start for radio frame *n.* A trigger signal is generated that signals the radio frame start to the transmit processing chain.
- This trigger is delayed by "**tx radio frame baseband delay**" clock cycles and then forwarded to the UL baseband transmitter (loop 2 in Figure 6-1) chain to prepare the time domain samples to be transmitted in radio frame *n+1.*
- The same trigger is delayed by "**tx radio frame trigger delay**" clock cycles and then forwarded to the transmit part of the digital frontend (loop 3 in Figure 6-1), to start transmitting the prepared time domain signal.

The trigger delays are calculated as follows:

$$tx\ radio\ frame\ baseband\ delay = tx\ radio\ frame\ trigger\ delay - TX\ BB\ processing\ delay$$

$$tx\ radio\ frame\ trigger\ delay = clock\ cycles\ per\ radio\ frame - RX\ RF\ and\ Sync\ delay - TX\ RF\ delay$$

The delay mechanism in the digital frontend which is based on the **tx radio frame trigger delay** value, requires values in in terms of the actual sampling rate (that is, 120 MHz for 40 MHz BW USRP devices, 200 MHz otherwise).

Both delay values are determined using calibration routines. The calibration principle is as follows

- **tx radio frame trigger delay:**
  This delay is determined closed-loop by using a base station with calibrated transmit and receive paths. The base station starts transmitting synchronization sequences, the mobile station synchronizes and responds on the uplink. The base station measures the deviation of the received uplink signal to the base station radio frame timing. The **tx radio frame trigger delay** value is adjusted until the uplink receive signal at the base station matches the base station downlink radio frame timing.
- **tx radio frame baseband delay:**
  This delay value is adjusted at the mobile station such that the transmit processing starts early enough and underflows at the input to the DAC are avoided. By choosing a smaller delay, the processing starts earlier, which prevents underflows since the baseband generates samples slightly faster than the RF loop consumes them.

7 Multi-Antenna Mobile Station FPGA Implementation

The multi-antenna mobile station implementation (see Figure 4-2) borrows from the base station implementation and adds some features of the single antenna mobile station. The main differences to the base station implementation are:

- One RRH subsystem with up to 12 antennas instead of multiple RRH subsystems.
- The multi-antenna uplink transmission does not use precoding but a direct mapping of spatial layers to antennas.
- Synchronization and tracking functionality is implemented, like the single-antenna mobile station.
- The transmit part of the DFE loop verifies at end of each radio frame whether the mobile station is still synchronized to the base station. If this is not the case, the transmit part of the DFE does not read from the FIFO which connects to the transmit I/Q processing loop and stops transmitting. The base station, on the other hand transmits continuously.

Sections 7.1 and 7.2 cover Inter-RRH synchronization and transmit-receive timing calibration. Synchronization and tracking, which are part of the RRH, are explained in section 7.3. Section 7.4 covers the layer-to-antenna mapping which is part of the MIMO processor FPGA.

7.1 Inter-RRH Time and Frequency Synchronization

Inter-RRH time and frequency synchronization are achieved similarly to the base station. Note that the receiver acts as frame timing master at the mobile station, as opposed to the base station where the transmitter acts as a frame timing master. That is, all RRHs receive a trigger signal like the base station. However, this trigger signal starts the signal

reception instead of the signal transmission. Also, this trigger starts a counter in the digital frontend simultaneously in all RRHs.

All RRHs are frequency-synchronous through a 10 MHz reference clock distributed to all RRHs.

7.2 Transmit-Receive Timing Calibration

The transmit-receive timing calibration is implemented similarly to the single-antenna mobile station (see section 6.3.4).

The transmit timing is determined in the following steps:

- The synchronization algorithm in the DL baseband receiver estimates the radio frame start for radio frame *n.* It forwards the respective DFE counter value to the DFE loop.
- Within the DFE loop a delay by "**tx radio frame trigger delay**" clock cycles is added to the DFE counter which corresponds to the radio frame start. Whenever the DFE counter equals this value, the transmit radio frame is started.

The trigger delays are calculated as follows:

$$tx\ radio\ frame\ trigger\ delay = clock\ cycles\ per\ radio\ frame - RX\ RF\ and\ Sync\ delay - TX\ RF\ delay$$

The delay mechanism in the digital frontend which is based on the **tx radio frame trigger delay** value, requires values in in units of the actual sampling rate (that is, 120 MHz for 40 MHz BW USRP devices, 200 MHz otherwise).

The delay value is determined using calibration routines. The calibration principle is as follows.

tx radio frame trigger delay:
This delay is determined closed-loop by using a base station with calibrated transmit and receive paths. The base station starts transmitting synchronization sequences, the mobile station synchronizes and responds on the uplink. The base station measures the deviation of the received uplink signal to the base station radio frame timing. The **tx radio frame trigger delay** value is adjusted until the uplink receive signal at the base station matches the base station downlink radio frame timing.

7.3 Multi-antenna Synchronization and Tracking

Figure 4-2 introduced the architecture of a multi-antenna mobile station. The first out of up to six RRHs comprises time and frequency synchronization functionality as well as tracking functionality like the single antenna mobile station. The first RRH also shares the synchronization results with the other RRHs. All RRHs apply the synchronization results synchronously.

Figure 7-1 shows the implementation block diagram of the first RRH (RRH 0, Combiner), which includes synchronization functionality. In addition, the time domain signal received at two antennas, once synchronized, is converted to frequency domain and the used subcarriers are extracted, like the RRH functionality of a base station. Each RRH forwards the synchronized frequency domain receive signal to the MIMO processor.

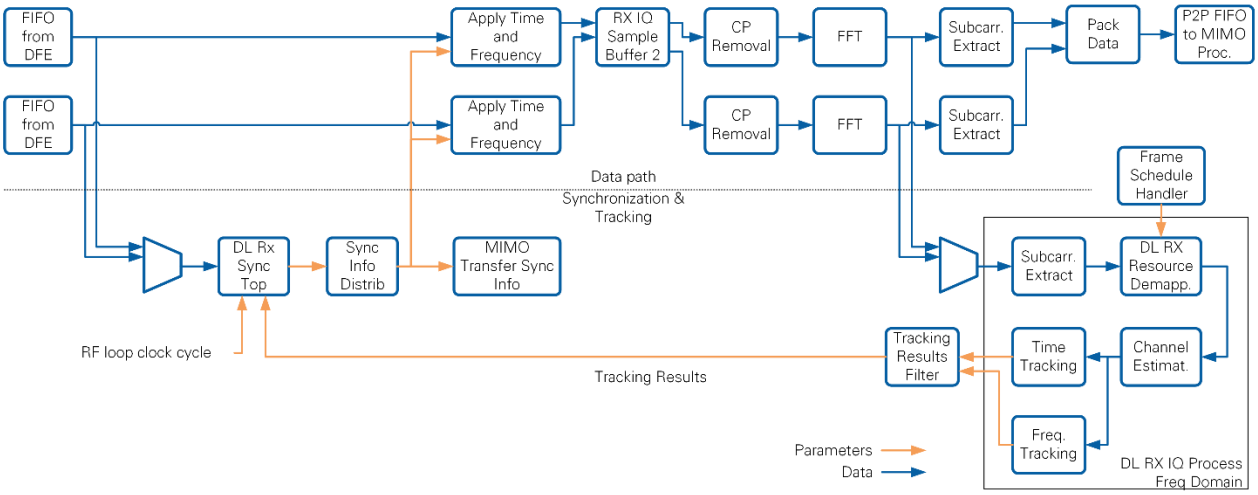

Figure 7-1. Multi-Antenna Mobile Station Receive Synchronization Implementation

The detailed functionality of the different RRH signal processing blocks shown in Figure 7-1 is as follows:

- Unsynchronized time domain data is received through P2P FIFOs from the two receive digital frontends which implement mainly re-sampling.
  It can be selected which of the two receive signals is used for synchronization. The selected signal is fed into the **DL Rx Sync Top** block (**Downlink Receive Synchronization Top Level**). This block does also continuously receive a counter value from the DFE loop (RF loop clock cycle). This block implements initial synchronization and merges these results with the tracking results which are fed back from subsequent modules. It comprises all functionality explained for the single antenna mobile station already (see Figure 6-3). The main difference is that this block is executed not as part of the data path, but in parallel. This block provides the index of the radio frame start, the carrier frequency offset, a Boolean signal whether the synchronization was found and the RF loop clock cycle which correspond to the radio frame start at its output (synchronization results). Those synchronization results become valid after the radio frame start index has passed on the parallel data path or on sync loss. This ensures that on the local RRH the results are not applied before they become available at other RRHs.
- The synchronization results are applied to the receive signal by the **Apply Time and Frequency** block, which is part of the data path**.**
- The synchronization results are fed into the **Sync. Info Distrib.** block. It has the following two functionalities:

    - RRH0: this block shares the synchronization results with other RRHs through a P2P FIFO. Also, it forwards the synchronization results to the Apply Time and Frequency block on RRH 0. Note that the synchronization result is forwarded from one RRH to the next RRH in a daisy chaining fashion instead of distributing it to all other RRHs directly from RRH0. Daisy chaining saves P2P FIFO resources at RRH0.
    - Other RRHs: this block reads the synchronization results computed at RRH0 and forwarded from RRH0 or another RRH from a P2P FIFO. Also, it forwards the synchronization result to the next RRH. Note that the last RRH in the subsystem does not further forward the synchronization results.
- The **MIMO Transfer Sync Info** block writes the synchronization results, forwarded by the Sync Info Distribution block, to registers. These registers are used for the following two purposes:
    - The uplink transmitter adjusts the uplink frame timing in the DFE loop. The transmitters in all RRHs will adapt the radio frame start accordingly and start transmitting simultaneously.
    - To correct the carrier frequency offset in the uplink transmitter.
- The time and frequency synchronized time domain signal is fed into the **RX IQ Sample Buffer 2** block. This block buffers incoming samples of both parallel data paths, aligns them with an OFDM symbols start signal and a sample valid signal, and delivers consecutive OFDM symbols to subsequent blocks.
- The **CP Removal** block, the **FFT** block and the **Subcarrier Extract** block implement the exact same functionality as introduced for the single antenna mobile station (see section 6.2.2). The Subcarrier Extract block provides the 1,200 used subcarriers per OFDM symbol at its output.
- The frequency domain subcarriers corresponding to two antennas are packaged by the **Pack Data** block as described in section 4.3.1 and forwarded to the MIMO processor through a P2P FIFO.
- The frequency domain signal, computed by the FFT block is also used for tracking purposes. Tracking parameters are computed by the **DL RX IQ Process Freq Domain** block, similar to the single antenna mobile station. The tracking functionality reuses blocks which are part of the single antenna mobile station frequency domain I/Q processing (**Subcarr Extract**, **DL RX Resource Demapp**, **Channel Estimat**., **Time Tracking**, **Frequency Tracking**). These have been introduced in section 6.2.2 (see Figure 6-4). The DL RX IQ Process Freq Domain block delivers the residual carrier frequency offset and the timing deviation from the first sample of a radio frame at its output (tracking results), once per radio frame.
- The **Tracking Results Filter** block forwards every second tracking result to the DL Rx Sync Top block which merges it with the initial synchronization results.

Use of tracking results in every second radio frame:
Tracking results are computed based on the first two downlink pilot OFDM symbols

received in a radio frame. These pilots can also be transmitted in the very end of a radio frame as shown in the example in Figure 7-2.

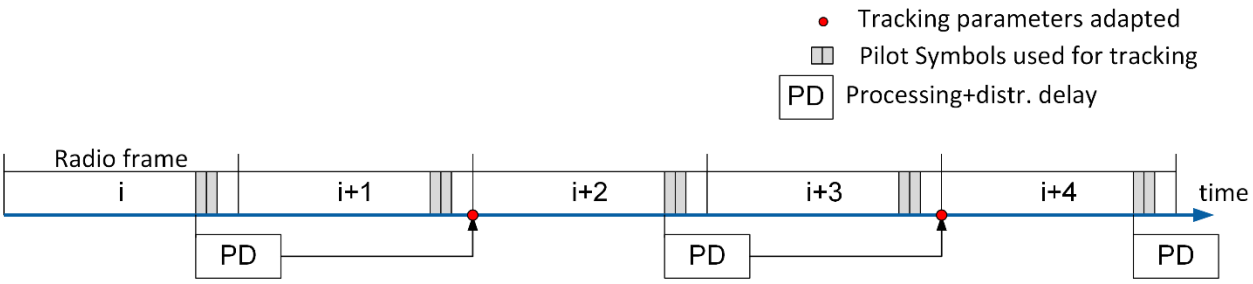

Figure 7-2. Multi-Antenna Mobile Station Synchronization Timing

In that case, it cannot be guaranteed that the tracking results will be available at the beginning of the next radio frame already. Hence, tracking results computed in radio frame i are applied at the beginning of radio frame $i + 2$. Using the tracking result from every radio frame with a delay of two radio frames can cause unstable synchronization behavior. By applying only every second tracking result it can be ensured that the correction according to the previous measurement has been applied before the next tracking results are computed.

7.4 Layer Mapping

The multi-antenna mobile station does not apply precoding in its uplink transmission but maps spatial layers directly to antennas as described in section 3.9.1. This functionality is part of the MIMO processor, similar to the base station.

The same MIMO processor implementation as described in section 5.2 is used. The difference is as follows:

- The base station computes MIMO equalization coefficients in the receive path of the MIMO processor as explained in section 5.2.3. These equalization coefficients are stored in a block RAM memory and used in the transmit path for precoding as described in section 5.2.4.
- The multi-antenna mobile station writes layer mapping matrices containing "1" and "0" values to the same block RAM in the same format as compared to the base station, instead of equalization coefficients. These layer mapping matrices are written from the host to the block RAM on run time. The layer mapping matrices are used for precoding instead of equalization weights. They are updated whenever the layer mapping is changed on the multi-antenna mobile station host. The equalizer weights in the receive path remain unchanged.

Assume, for instance, the multi-antenna mobile station employs four antennas. Layers (0,3) have been selected for transmission in uplink and downlink. The equalization operation is shown below for this example.

$$\begin{bmatrix} \hat{x}_0 \\ 0 \\ 0 \\ \hat{x}_3 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} W_{0,0} & W_{0,1} & W_{0,2} & W_{0,3} & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ W_{3,0} & W_{3,1} & W_{3,2} & W_{3,3} & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{7.1}$$

Note that the equalization matrix has a static dimension of 12 rows (maximum number of spatial layers) and 32 columns (maximal number of antennas). The eight bottom rows in the example in (7.1) are filled with zeros. The 28 right most columns in (7.1) are filled with zeros as well. The equalization operation yields a length-12 column vector which has non-zero elements only in positions (0,3), that is, the layers which have been chosen for transmission.

At the base station, the matrix $\boldsymbol{W}$ would be written into the block RAM (see section 5.2.3.9) and used for precoding. At the multi-antenna mobile station, the following matrix of dimension 12 x 32 is written to the block RAM instead

$$\begin{bmatrix} \mathbf{1} & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix}. \tag{7.2}$$

The transpose of this matrix is used during the precoding operation in the transmit path as shown below

$$\begin{bmatrix} \tilde{x}_0 \\ \tilde{x}_1 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ 0 \\ 0 \\ x_3 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{7.3}$$

Note that the signal vector $\boldsymbol{x}$ has a static length of 12 according to the maximum number of layers while the transmitted signal vector $\tilde{\boldsymbol{x}}$, after layer mapping, has a length of 32 according to the maximum number of antennas. Spatial layer 0 is mapped to antenna 0 and spatial layer 3 to antenna 1.

Note that the process of writing layer mapping matrices from the host to the FPGA block RAM may take multiple radio frames. The multi-buffer concept explained in the MIMO processor section 5.2.3.9 guarantees that the same layer mapping matrices are used for all RBs within one radio frame, even if the layer mapping matrices are updated.

8 Host Implementation

8.1 Overview

The host implementation orchestrates the interaction between all FPGA components. It initializes all FPGAs and sets up the interconnection between all components. It furthermore sets parameters during runtime and queries and displays the system status. It also acts as the connection to the lower MAC interface on the FPGA where it supplies payload data to the TX and collects payload from the RX for further processing on the host.

This chapter describes the host code architecture with focus on the base station implementation. Important differences to the mobile stations will be noted.

8.2 Performance Considerations

Data between host and FPGA are exchanged either by DMA FIFOs or in terms of controls and indicators. Because of the non-real time execution nature of host code, it must be ensured that FIFOs are large enough to compensate host execution time jitter. Note that the host might miss fast state changes signaled by FPGA indicators, as these are queried with a periodicity of multiple 100 ms but may change on a much faster time scale.

LabVIEW distributes different while loops running on the host to all available CPU cores. If there are more loops on the host top level than cores available on the CPU, the system starts swapping the running loops between the cores which results in poor performance. Therefore, it is a recommended practice to keep the number of loops below the number of logical cores (eight for a PXIe-8135 module).

8.3 Front Panel Layout

All front panels are laid out in a similar way as shown in Figure 8-1. The bar on the top contains the title, some minimal information and instruction about the purpose of the host on the left, and some basic health and performance indicators as well as a stop button on the right.

The left-hand side contains the most important controls which must be adjusted by the user before starting the system.

The main part of the front panel (center and right) provides access to detailed configuration settings and monitoring parameters distributed over multiple tabs. Note that a detailed description of all controls and indicators is available in *MIMO Application Framework Getting Started Guide* [2].

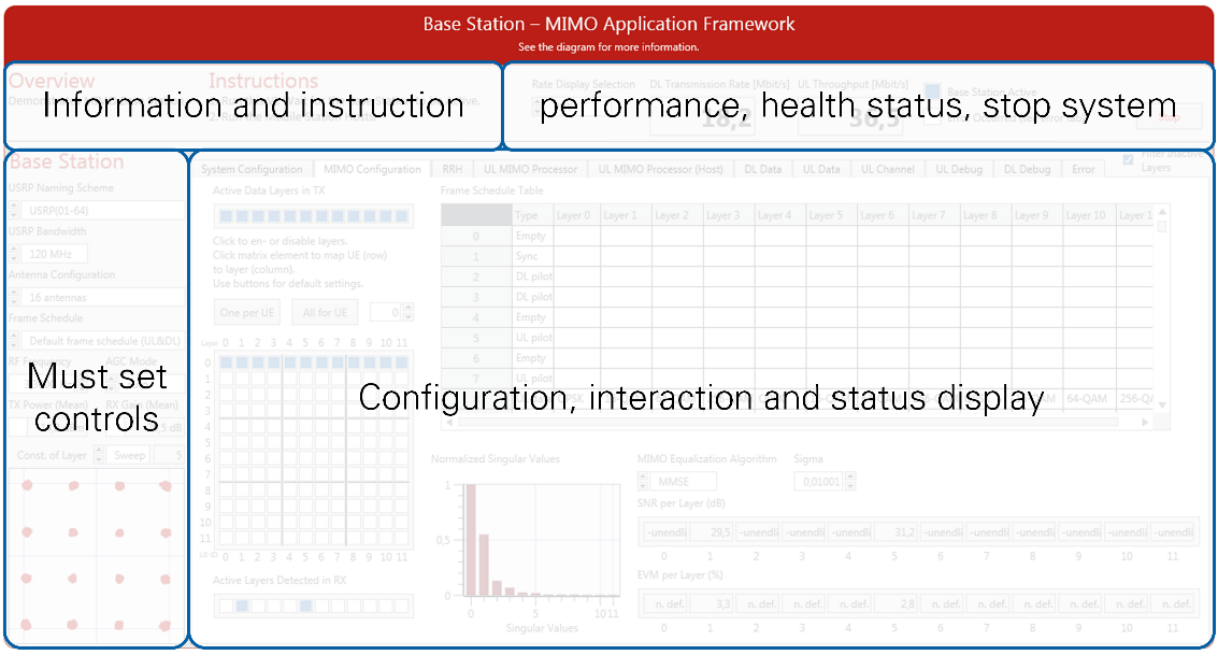

Figure 8-1 General Front Panel Layout

8.4 Block Diagram Layout

All host block diagrams follow a basic layout scheme shown in Figure 8-2. On the left, there is the initialization code needed for the system. In the center of the VI there are several loops which are executed continuously on runtime. These loops configure the FPGA, display the system status and send and receive payload data. On the right-hand side, there is the cleanup code which is executed on system shutdown.

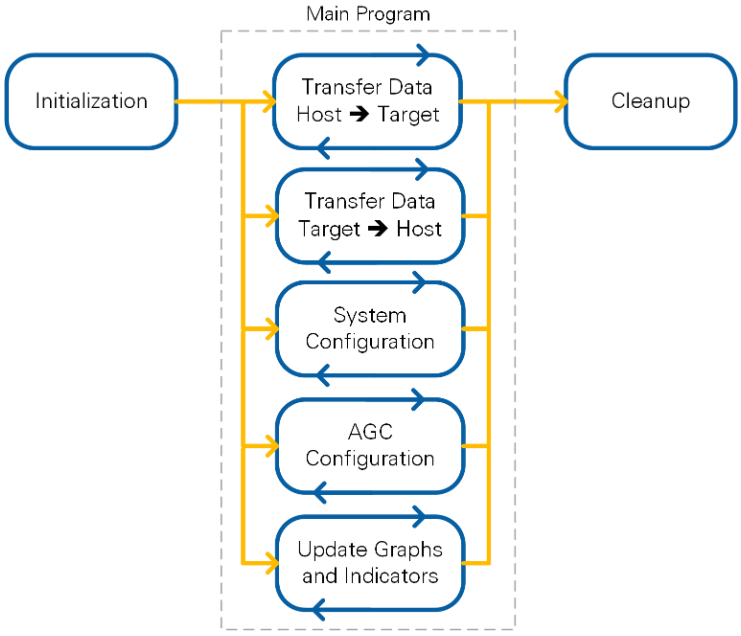

Figure 8-2 General Block Diagram Layout

The host executes sequentially as indicated by the yellow arrows in the following order

1) Initialization → 2) Main Program → 3) Cleanup.

Handles that are persistent at runtime (such as FPGA interfaces, UDP port handles, or FIFOs) are grouped in a session cluster to minimize the number of wires on the block diagram.

8.5 Initialization

Initialization starts by resetting all controls and indicators in a sequence structure to initialize all controls and indicators to a known state. This ensures proper execution also after a previous run.

The startup procedure at the **base station** is as follows:

1) Initialize the system.
   a) Open UDP ports.
   b) Check user defined configuration.
   c) Load bitfiles.
   d) Initialize USRP transceivers, for instance carrier frequency and sampling rate.
   e) Start DMA FIFOs.
   f) Configure FPGAs, for instance the router program, the radio frame schedule or the user-to-layer mapping.
   g) Configure P2P FIFO connections (see Figure 4-4 for an example).
   h) Reciprocity calibration. See section 3.6 for the procedure. Note that reciprocity calibration includes a waiting time to settle the system at operating temperature first.
2) Check for Initialization Errors: generates detailed error messages.
3) Start the transmission.
   a) Start Bit Processor and MIMO processor.
   b) Generation of transport blocks on the bit processor for all 12 data sources. These transport blocks contain padding only as no payload is yet available. Note that the respective frequency domain signal is set to zero at the MIMO processor as no layers are active yet.
   c) This "zero signal" is fed into the RRHs. A PSS is inserted per radio frame. Once the transmit waveform which contains zeros except for the PSS OFDM symbol is available at the input to the DAC of each RRH, the transmission is started.
   d) Start of the transmission is issued from the host. RRH0 will generate a trigger signal which is distributed to all RRHs including RRH0. Upon reception of this trigger, all RRHs start transmitting simultaneously.

The following aspects are different at the mobile stations.

Multi-antenna mobile station

The initialization procedure is similar to the base station except for the reciprocity calibration which is not implemented at the mobile station. At the mobile station the signal reception is started synchronously instead of the signal transmission, through the same trigger procedure as compared to the base station.

Single-antenna mobile station

Two single-antenna mobile stations are implemented on a single USRP device. Both are initialized in parallel.

The general execution order of initialization steps is similar as compared to the base station. However, many steps executed for the base station are not required, such as initialization of P2P connections or reciprocity calibration.

8.6 Main Program

The main program executes in multiple parallel while loops. The following functionality is implemented on the host.

1) Payload data handling
   a) Read UDP packets or generate PN data.
   b) Format payload into DTP packets and send these packets to the bit processor FPGA.
   c) Read receive indications from the bit processor FPGA, extract payload, and forward to respective UDP ports.
   d) Evaluate and display throughput and block error rates.
2) Monitoring of transmission parameters
   a) Read OFDM symbols from the MIMO processor FPGAs.
   b) Compute and display channel transfer functions and impulse responses.
   c) Equalize received data at the host and display constellations. Compare to constellations computed at the FPGA.
3) Change run-time configurable FPGA parameters (for example, MIMO equalization algorithm).
4) Compute receive gains (AGC at mobile station and base station) and uplink power control parameters (mobile station only) (see section 3.8).
5) Detect active layers at the base station (see section 3.9).
6) FPGA interface monitoring: monitor FIFO overflows, throughput on interfaces.

8.7 Stop Procedure and Cleanup

The host must ensure a proper shutdown. Especially, all handles to resources (hardware and software) allocated by the host must be closed.

8.7.1 Stop Procedure

All loops need to be stopped before the cleanup code is executed. Three conditions can stop a loop:

1) An error happened in the loop.
2) The stop button has been pressed. Note that the stop button is monitored in a single loop.
3) Another loop has stopped for condition 1) or 2).

Conditions 1 and 2 only stop a single loop. Other loops must be notified to stop as well. A notifier is used for this purpose. The notifier name is customized using the filename of the host. The loop which stopped first creates a notifier using the `Stop on Error.gvi`. This notifier is received by all other loops using `Check Stop.gvi`. Upon receiving the notifier, the other loops will stop as well. See Figure 8-3 for a typical use of both nodes.

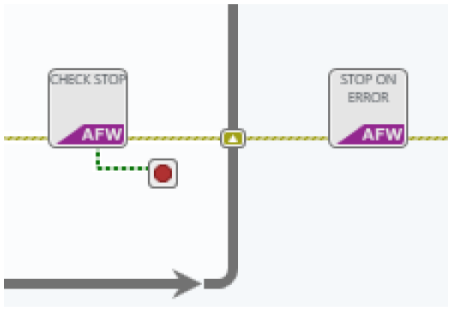

Figure 8-3. Use of the Stop Notifier in Host Loops

Users need to ensure to follow this design pattern when adding new loops to the hosts.

8.7.2 Cleanup Procedure

The cleanup procedure is executed in two steps.

First, all handles related to data handling such as UDP handles and queue handles are closed as shown in Figure 8-4. Second, hardware related handles such as P2P handles or FPGA references, are closed.

The host uses the information from the session cluster to identify resources to be deallocated.

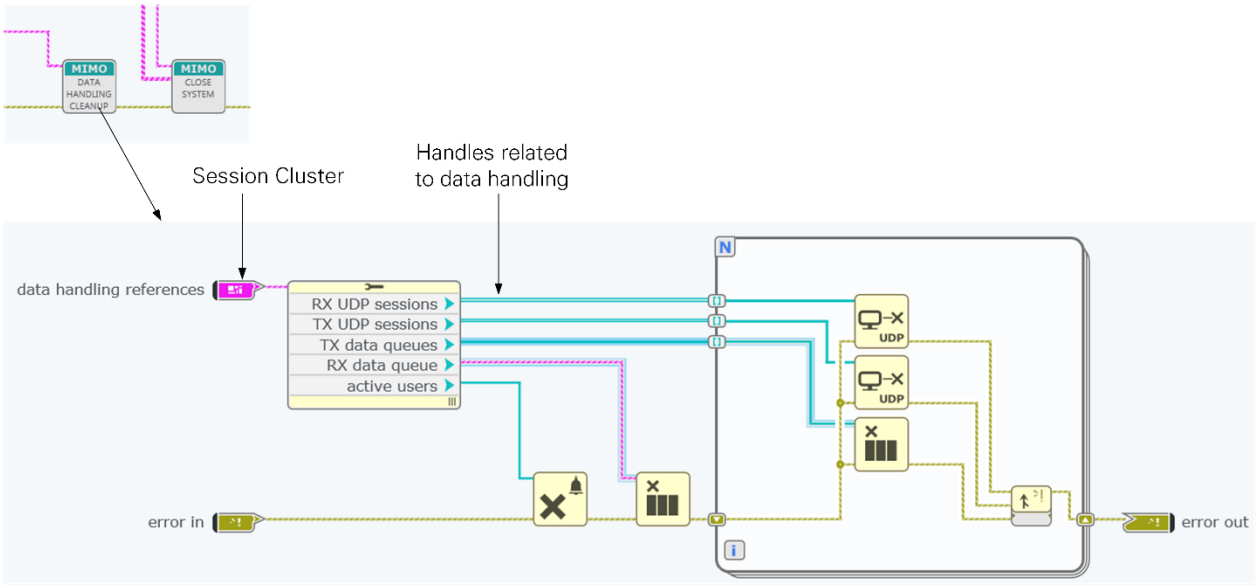

Figure 8-4. Closing Data Handling Related Resources on the Host

9 Example Experimentation Results

9.1 Introduction

This chapter summarizes example experimentation results obtained indoors comparing single-user MIMO and multi-user MIMO transmission. These results should be interpreted as examples for the specific propagation conditions and the chosen antenna element orientations. Different distances, antenna arrays and propagation environments will yield different results.

Section 9.2 summarizes the experimentation setup. Section 9.3 presents the experimentation results.

9.2 Experimentation Setup

The experiments have been carried out indoors. Figure 9-1 illustrates the room layout and the base station and mobile station positions. Different base station hardware configurations are placed on the left and different mobile station configurations on the right. The distance between base stations and mobile stations is approximately 5 m. Note that the three individual base stations can be combined into a 128-antenna base station if needed for the experiment.

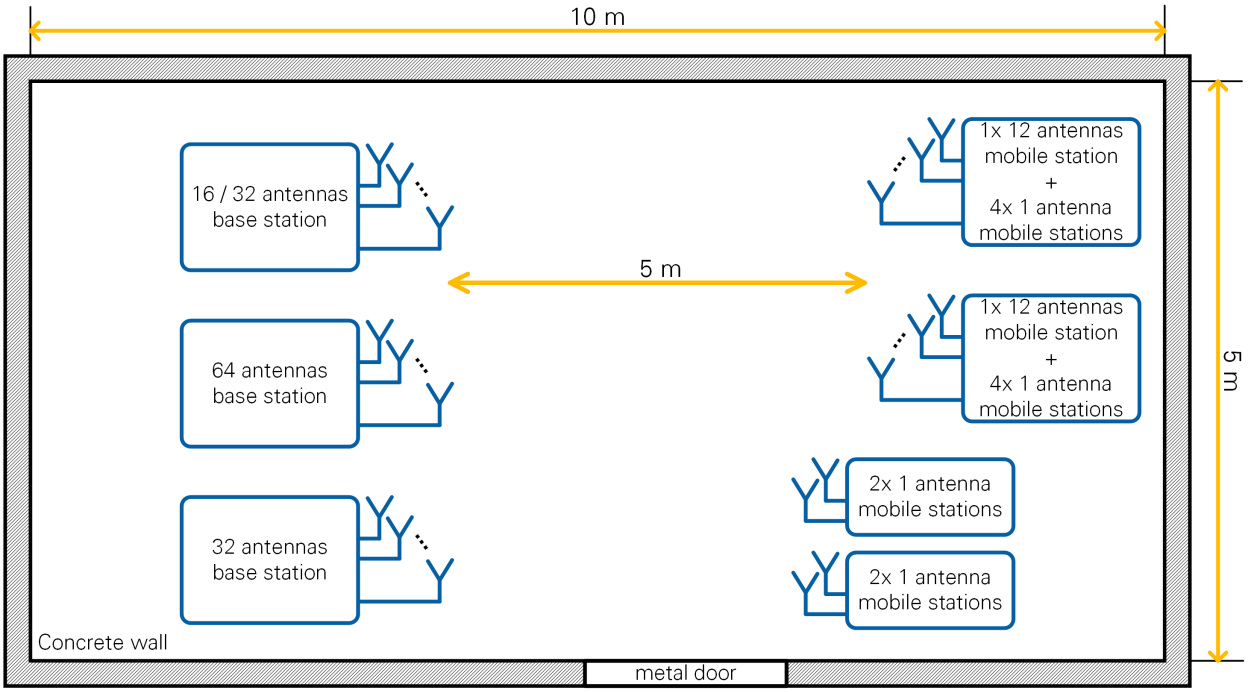

Figure 9-1. Experimentation Room Layout

Note that standard dipole antennas are directly attached to the USRP devices.
The experiments have been carried out at a carrier frequency of 3.61 GHz. The radio frame schedule was configured such that data symbols which immediately follow a pilot symbol are evaluated. Two uplink data symbols and one downlink data symbol are scheduled per slot. All tests used MMSE based equalization and precoding.

9.3 Results

A primary goal of the tests was to identify how many spatial layers can be supported at a certain modulation scheme with different antenna configurations.

Test condition: Layers are declared to be correctly received using a certain modulation scheme if the block error rate remains below 0.01 for several minutes. Note that this is a very strict criterion as no error correction coding is employed and a single bit error will cause a block error.

For throughput and block error rate measurement purposes, it was ensured that the full buffer assumption was valid. That is, the PN source in UL and DL generated data a rate more than the maximum rate supported by each setup.

9.3.1 Case 1: Single User MIMO Performance with 128 Base Station Antennas

This test evaluates the maximum single-user MIMO performance using a 128-antenna base station and a 12-antenna mobile station. Example received constellations are shown in Figure 9-2 and Figure 9-4. It was found that 12 layers could be transmitted at 64-QAM modulation in uplink and downlink. The post equalization SNR was about 30 dB at each layer in uplink and downlink.

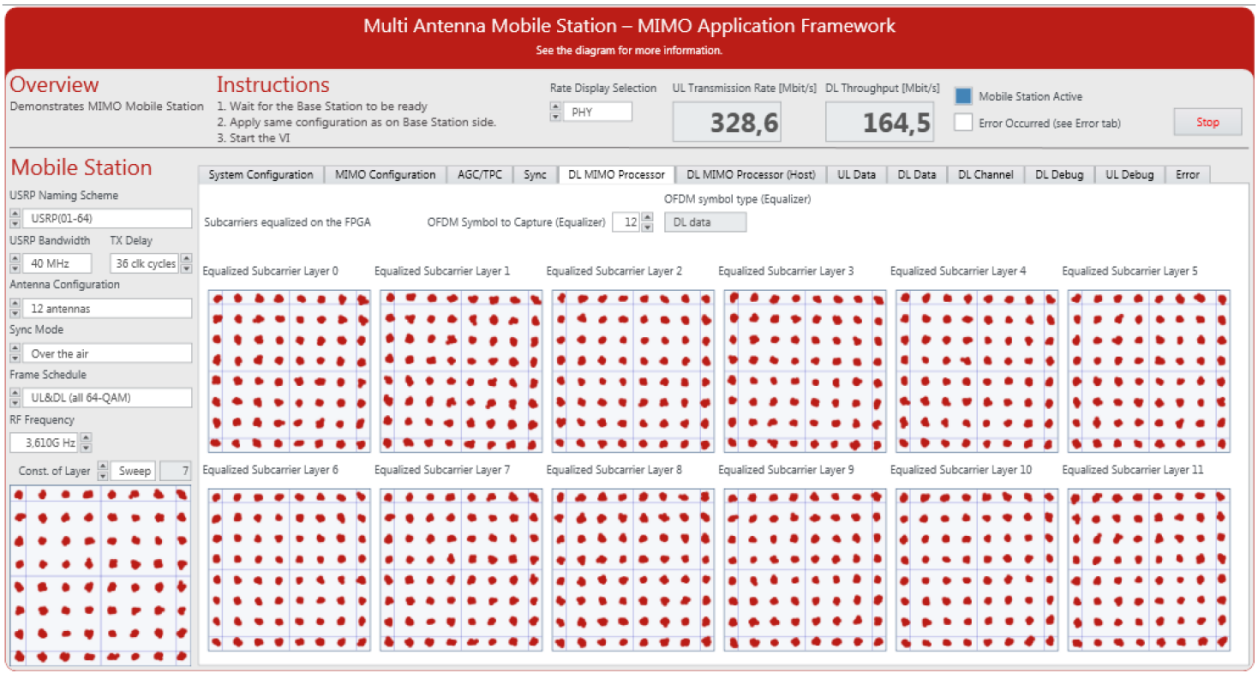

Figure 9-2. Received Constellation at the Multi-antenna Mobile Station (128 x 12)

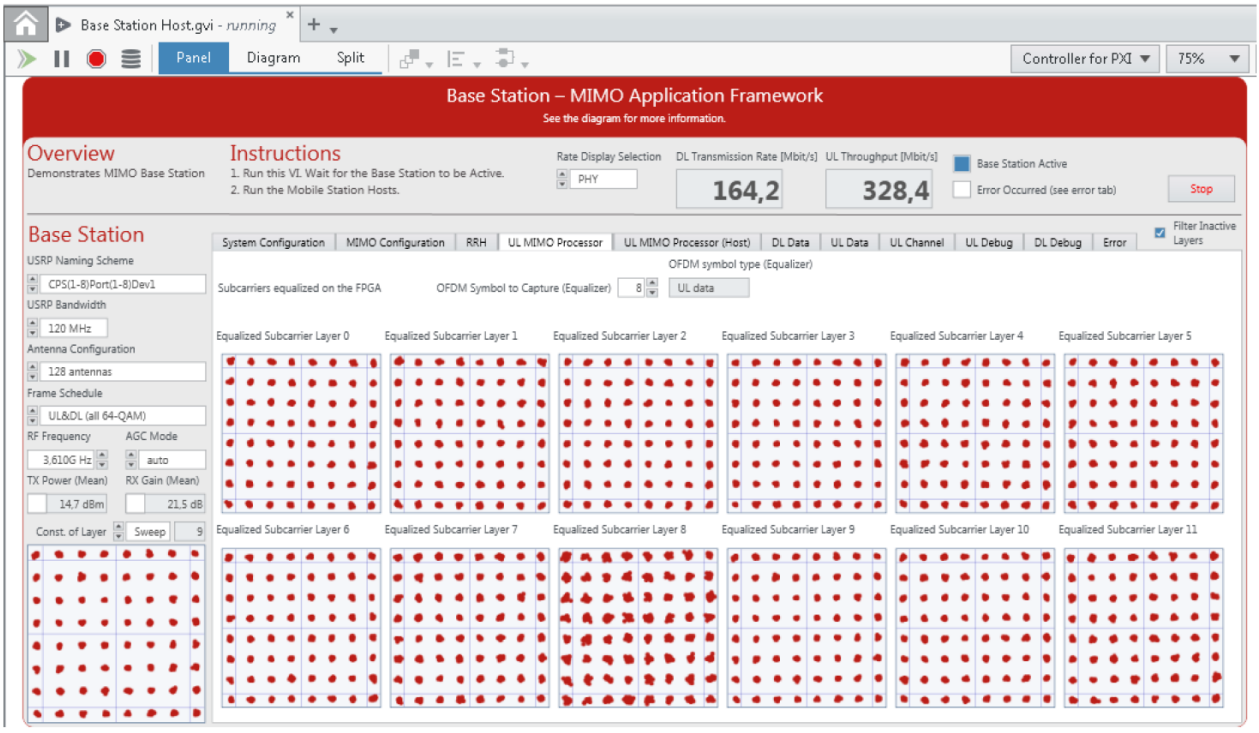

Figure 9-3. Received constellation at the base station (128 x 12).

The underlying singular values of the MIMO channel for uplink and downlink channels are shown in Figure 9-4.

Figure 9-4. MIMO channel singular values at the mobile station (left) and at the base station (right).

Note that the singular values of the downlink channels are computed based on the effective channel including precoding. Therefore, the singular values of the downlink differ from the singular values of the uplink channel, which have been computed from the channel estimate without precoding.

9.3.2 Case 2: Single-User MIMO Performance @ 64 Base Station Antennas

This test evaluates the single-user MIMO performance using a 64-antenna base station and a 12-antenna mobile station.

Example received constellations are shown in Figure 9-5 and Figure 9-6. It was found that 11 layers could be transmitted at 64-QAM modulation. That is, the base station with 128 antennas tested in the previous subsection could support one additional layer.

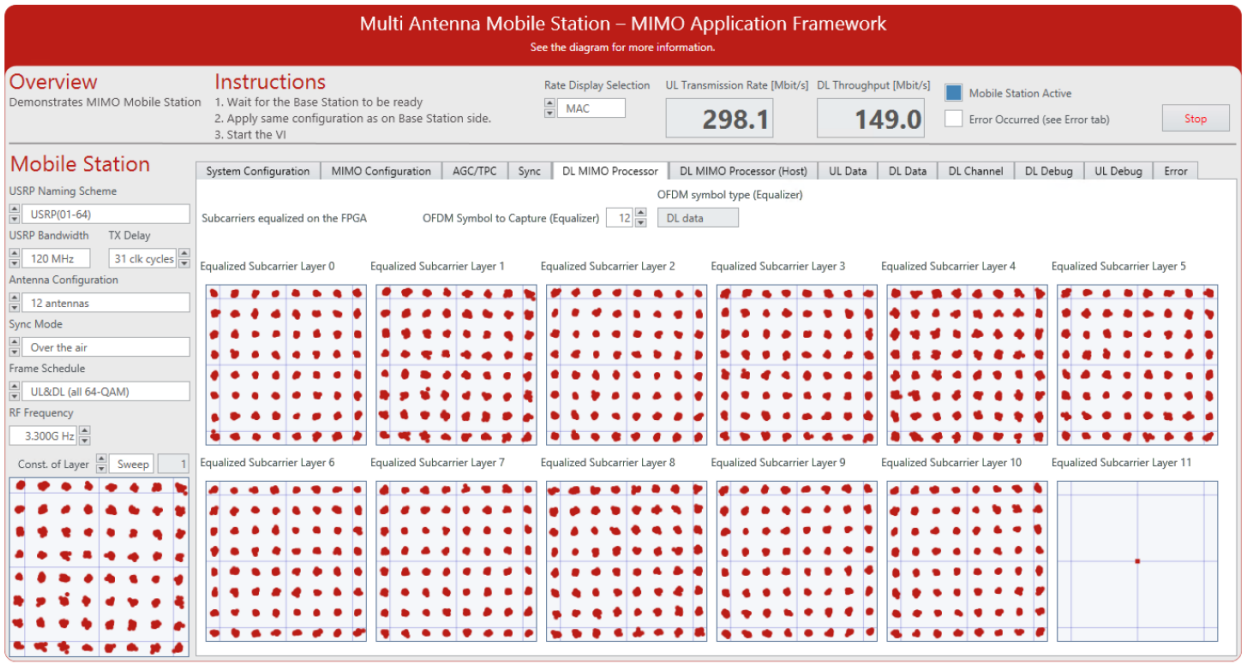

Figure 9-5. Received constellation at the multi-antenna mobile station (64 x 11).

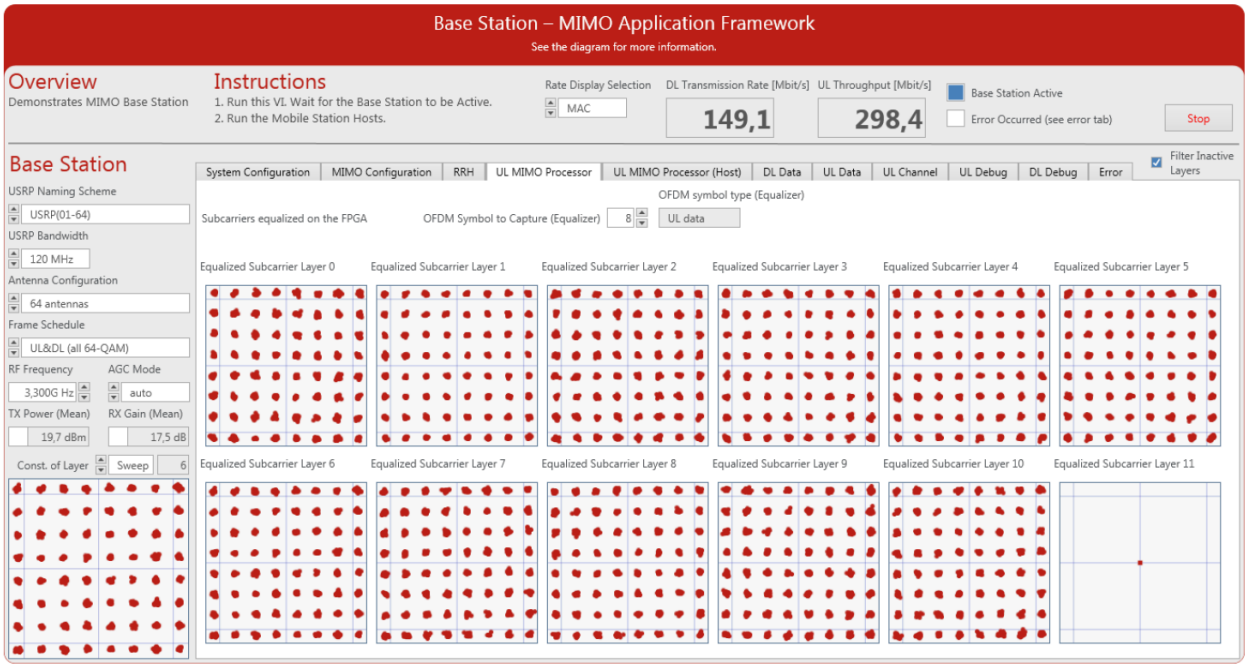

Figure 9-6. Received constellation at the multi-antenna mobile station (64 x 11).

9.3.3 Case 3: Multiuser MIMO Performance @ 64 Base Station Antennas

This test evaluates the single-user MIMO performance using a 64-antenna base station and multiple single-antenna mobile stations.

Results are shown in Figure 9-7 and Figure 9-8. The performance-limiting factor, in this case is the downlink where some residual spatial interference remains which cannot be suppressed at the single antenna mobile stations. Eight spatial layers can be supported in this test at 0.01 block error rate.

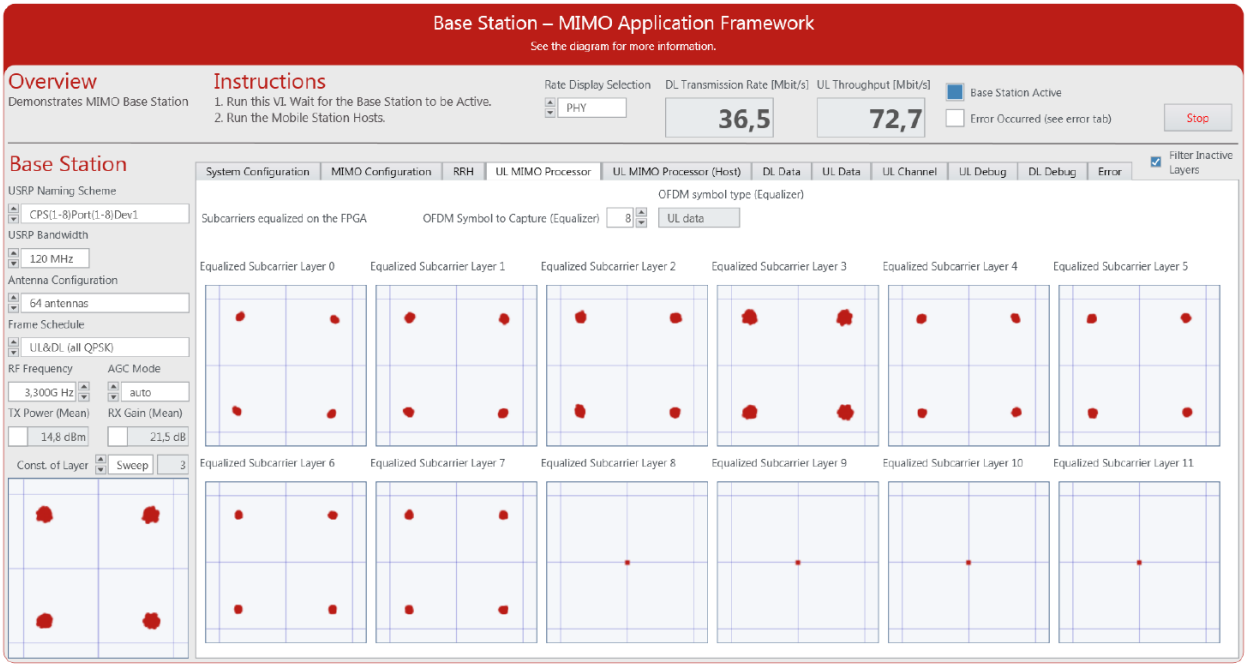

Figure 9-7. Received Constellation at the Base Station (64 x 8)

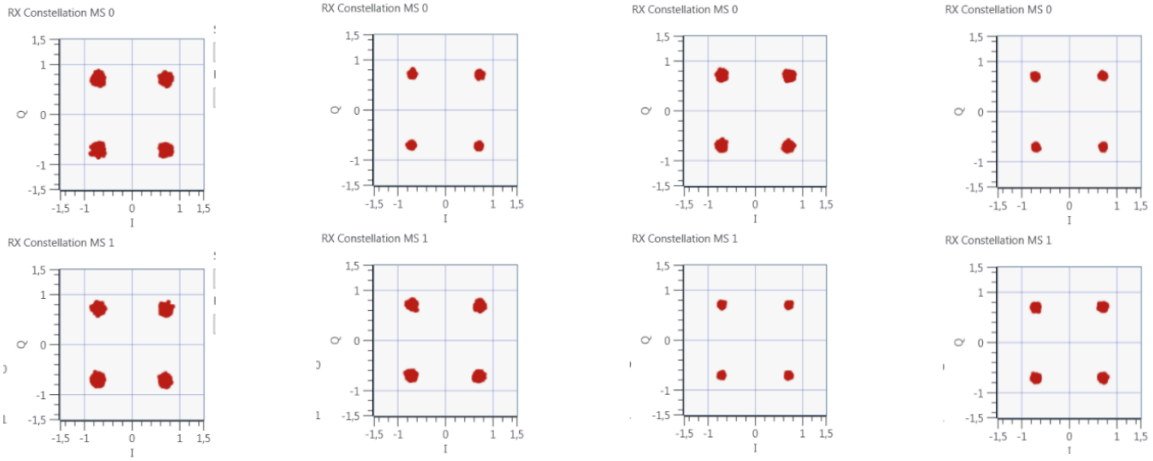

Figure 9-8. Received Constellation at Eight Single Antenna Mobile Stations (64 x 8)

9.3.4 Experimentation Summary

A variety of base station antenna configurations have been tested, ranging from 128 antennas to two antennas. Results for the maximum number of layers that can be supported at 0.01 block error rate are reported in Table 9-1 for reference.

Note that there are multiple possible ways to configure a 32-antenna base station, for instance, using the large setup shown in Table 9-1. Each configuration will result in using RRHs placed at different positions within the room and using different antenna orientations. The results in Table 9-1 are valid only for a single configuration and may change completely in case a different hardware configuration is chosen. It is important to interpret these results as examples which heavily depend on the specific hardware configuration and the propagation conditions.

Table 9-1. Supported Spatial Layers and Modulation Schemes for Different Antenna Combinations

System	Uplink	Downlink
128-ant BS + 1 x 12-ant MS	12 layers with 64-QAM	12 layers with 64-QAM
128-ant BS + 8 x SISO MSs	8 layers with 64-QAM	8 layers with 64-QAM
64-ant BS + 1 x 12-ant MS	12 layers with 64-QAM	12 layers with 16-QAM 11 layers with 64-QAM
64-ant BS + 1 x 8-ant MS	8 layers with 64-QAM	8 layers with 64-QAM
64-ant BS + 8 x SISO MSs	8 layers with 64-QAM	8 layers with QPSK 4 layers with 16-QAM
32-ant BS + 1 x 8-ant MS	8 layers with 16-QAM 7 layers with 64-QAM	8 layers with 16-QAM 6 layers with 64-QAM
32-ant BS + 8 x SISO MSs	8 layers with 64-QAM	8 layers with QPSK 5 layers with 16-QAM
16-ant BS + 1 x 4-ant MS	4 layers with 64-QAM	4 layers with 64-QAM
16-ant BS + 4 x SISO MSs	4 layers with 64-QAM	4 layers with 16-QAM 2 layers with 64-QAM + 2 layers with 16-QAM
12-ant BS + 1 x 12-ant MS	11 Layers with QPSK 8 Layers with 16-QAM 5 Layers with 64-QAM	9 Layers with QPSK 7 Layers with 16-QAM 3 Layers w/ 64-QAM
8-ant BS + 1 x 8-ant MS	8 Layers with QPSK 6 Layers with 16-QAM 5 Layers with 64-QAM	7 Layers with QPSK 5 Layers with 16-QAM 3 Layers with 64-QAM
4-ant BS + 1 x 4-ant MS	4 Layers with QPSK 4 Layers with 16-QAM 2 Layers with 64-QAM	4 Layers with QPSK 3 Layers with 16-QAM 1 Layer with 64-QAM
2-ant BS + 1 x 2-ant MS	2 Layers with QPSK 2 Layers with 16-QAM 1 Layer with 64-QAM	2 Layers with QPSK 1 Layer with 16-QAM

10 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.

10.1 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.

10.2 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 80. The namespace layout is independent of the component layout; two components can share the same root namespace as shown in Figure 81.

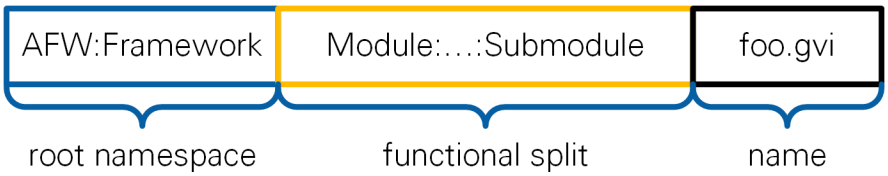

Figure 80 Namespace Hierarchy

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.

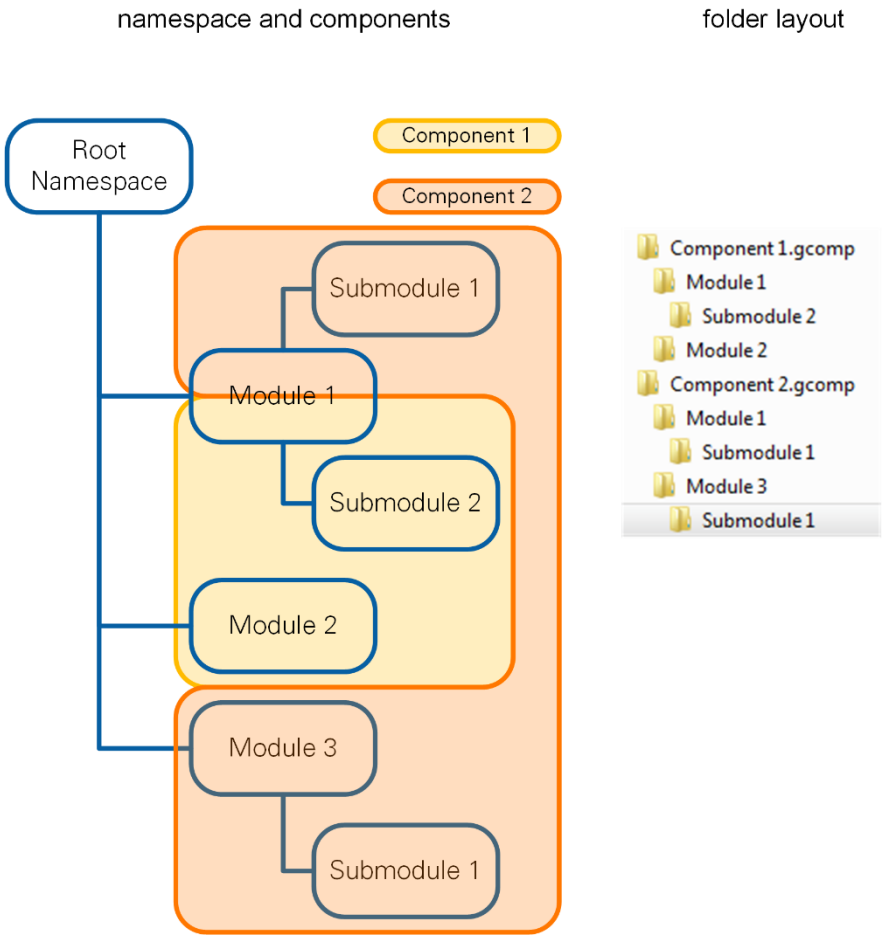

Figure 9 Connection between Namespaces, Components and Folder Layout on Disk

10.3 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.

11 Conclusions

NI technology revolutionizes the prototyping of high-end research systems with LabVIEW Communications System Design Suite software coupled with USRP and PXI hardware platforms. This manual details the capabilities of the application framework in combination with NI hardware to form the world's first working real-time Massive MIMO solution. The flexible MIMO prototyping system utilizes a unique combination of technology that enables the synchronization of time and frequency over 128 antennas, all using commercial off-the-shelf components. The application framework also uses design flows for the FPGA to simplify high-performance processing on the PHY and MAC layers to meet real-time timing requirements.

The application framework provides a comprehensive set of features that are presented in chapter 2. The system architecture allows to extend the application framework towards many features. The implementation details presented in chapters 5 through 8 give an overview to start with the existing system, edit and add new modules, or build your own system by reusing the existing modules and submodules. The FPGA files shipped with the design allow for prototyping including RF.

The application framework enables developers and researchers around the world to rapidly implement MIMO applications. They can save time and effort for new ideas development without dealing with data communication issues, data synchronization, system calibration, right hardware selection, and system compatibility. It is intended as a starting point for the implementation and validation of new systems, algorithms, and a variety of features such as the following:

- MAC extensions
- Instantaneous and statistical reciprocity
- Precoding design
- Space-time coding design
- Scheduling
- Secure massive MIMO
- Pilot allocation optimization
- Antenna array design
- Beamforming, with the possibility to focus transmitted signal energy into very small areas to provide huge improvements in capacity and study distributed massive MIMO systems
- Channel modeling
- High mobility
- UL interference management
- Multi-cell multiuser massive MIMO
- Prototype advanced receiver algorithms at both ends of communication link, for example, synchronization, channel estimation, and equalization

Because of the flexibility of LabVIEW Communications System Design Suite and the modularity of the framework, you can easily exchange portions of the design for prototyping new algorithms for future wireless systems. In addition, the native interface between the host and the FPGA in LabVIEW Communications System Design Suite means that the design can be partitioned to profit from the parallel execution on the FPGA as well as calculations on the host. With LabVIEW Communications System Design Suite, users can integrate additional programming approaches such as VHDL through the IP integration node and even .m file scripts through the LabVIEW MathScript RT Module. This makes it possible to develop high-performance implementations that are also highly readable and customizable.

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.

12 Bibliography

[1] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Transactions on Wireless Communications,* p. 3590–3600, 2010.

[2] "MIMO Prototyping System Getting Started Guide," National Instruments, 2016.

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

[4] IEEE, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 2012.

[5] D. Wubben, R. Bohnke, V. Kuhn and K.-D. Kammeyer, "MMSE extension of V-BLAST based on sorted QR decomposition," in *VTC Spring 2003*, Orlando, 2003.

[6] Y. Rao, "Implementing modified QR decomposition in hardware". Patent US9201849 B2, 30 Jan. 2013.

[7] C. Shepard, H. Yu, N. Anand, L. E. Li, T. Marzetta, R. Yang and L. Zhong, "Argos: Practical Many-Antenna Base Stations," in *MobiCom'12*, Istanbul, Turkey, 2012.

[8] N. Khoolenjani and K. Khorshidian, "Distribution of the Ratio of Normal and Rice Random Variables," *Journal of Modern Applied Statistical Methods,* vol. 12, no. 2, pp. 436-449, 2013.

[9] E. Ohlmer, T. J. Liang and G. F. Fettweis, "Algorithm for Detecting the Number of Transmit Antennas in MIMO-OFDM Systems," in *VTC Spring 2008*, Signapore, 2008.

[10] A. Gaber and A. Omar, "Utilization of Multiple-Antenna Multicarrier Systems and NLOS Mitigation for Accurate Wireless Indoor Positioning," *in IEEE Transactions on Wireless Communications,* vol. 15, no. 10, pp. 6570-6584, 2016.

[11] Xilinx, "Fast Fourier Transform v9.0 - LogiCORE IP Product Guide," 2015.

[12] 3. G. P. Project, "3GPP Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Access Network (E-UTRA); Base Station Radio Transmission and Reception (Release 11)," 2014.

[13] J. Vieira, K. Nieman, Z. Miers, . N. Kundargi, L. Liu, I. Wong, V. Öwall, O. Edfors and T. Fredrik, "A flexible 100-antenna testbed for Massive MIMO," in *IEEE Globecom Workshops (GC Wkshps)*, Austin, TX, 2014.

[14] J. Vieira, F. Rusek and F. Tufvesson, "Reciprocity calibration methods for Massive MIMO based on," in *IEEE Globecom 2014*, Austin, Texas, United States, 2014.

[15] J. Vieira, F. Rusek, O. Edfors, S. Malkowsky, . L. Liu and F. Tufvesson, "Reciprocity Calibration for Massive MIMO: Proposal, Modeling and Validation," *IEEE Transactions on Wireless Communications (submitted),* 2016.

[16] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors and F. Tufvesson, "Scaling up MIMO: Opportunities and Challenges with Very Large Arrays," *Signal Processing Magazine, IEEE,* p. 40–60, 2013.

377799C-01 Nov 2019