Wireless Innovation Forum Contribution

Committee: Coordinating Committee for International SCA Standards

Title: Specification of the OBISS IQ Baseband Interface

Source: Gerald Ulbricht, Gerd Kilian
Fraunhofer Institute for Integrated Circuits
Am Wolfsmantel 33, Erlangen 91058, Germany
+49 9131 776 3150
Gerald.ulbricht@iis.fraunhofer.de

Date: 20 November 2015
Distribution: Unrestricted

Document Summary: This specification defines the IQ baseband interface between a baseband unit and a transceiver module of a communication device. The interface enables the bi-directional transmission of digital IQ baseband signals as well as the transmission of control and management information.

The digital interface allows a transmission of IQ samples with rates between 1.2 and 72 MSamples per second. The interface transmits the samples at four different, predefined sample rates between 9.6 and 72 MSamples per second. Thus, waveforms with a bandwidth of up to 57 MHz can be transmitted.

Moreover, the specification supports a division of the IQ samples into up to eight transmitters or receivers in the transceiver module, for example in multiple antenna systems. In this case, the sample rate per transmitter or receiver is reduced, and also the maximum bandwidth for the waveform has been reduced in proportion to the number of transmitters and receivers addressed.

A survey of the transmission modes of the interface is given in Table 6 2 in Chapter 6.3.1.

In the baseband unit, IQ samples transmitted or received are allocated to specific points in time with the assumption of a fixed delay time for the sample rate set by the transceiver module. All other points in time in the transceiver module are referred to in reference to IQ samples.

TERMS, CONDITIONS & NOTICES

This document has been prepared by the The Fraunhofer Institute to assist The Software Defined Radio Forum Inc. (or its successors or assigns, hereafter "the Forum"). It may be amended or withdrawn at a later time and it is not binding on any member of the Forum or of the The Fraunhofer Institute.

Contributors to this document that have submitted copyrighted materials (the Submission) to the Forum for use in this document retain copyright ownership of their original work, while at the same time granting the Forum a non-exclusive, irrevocable, worldwide, perpetual, royalty-free license under the Submitter's copyrights in the Submission to reproduce, distribute, publish, display, perform, and create derivative works of the Submission based on that original work for the purpose of developing this document under the Forum's own copyright.

Permission is granted to the Forum's participants to copy any portion of this document for legitimate purposes of the Forum. Copying for monetary gain or for other non-Forum related purposes is prohibited.

THIS DOCUMENT IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NON-INFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY USE OF THIS SPECIFICATION SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER THE FORUM, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER, DIRECTLY OR INDIRECTLY, ARISING FROM THE USE OF THIS DOCUMENT.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the specification set forth in this document, and to provide supporting documentation.

Specification of the OBISS IQ Baseband Interface

TRXIX-IIS-FDSB-600-100000

Edition02 Revision00

Written by: Gerald Ulbricht, Gerd Kilian
Date: 23. November 2015

Name	Function	Date	Signature
Checked by: Gerd Kilian			
Released by: Gerald Ulbricht	Project manager		

Revision Index:

Ed./ Rev.	Date	Authorized by	Remarks	Author
01.00	26 Jan 09		First revision	ubt, kil
02.00	20 Nov 15		Amendment of command structure for time-critical and time-uncritical information Amendment at header: 1-bit flow control of TU package, 1 bit for alert flag, event flag for signaling an event Revision of acknowledge mechanism for TC and TU commands Details with regard to the time offset of the TX/RX message amended Frame synchronization modified CRC error flag and acknowledge for time-critical commands amended Modification of the SOM definition Revision of Chapter "Time Synchronization" Annex	ubt, kil, kln

1 Table of Contents

2 Abstract

This specification defines the IQ baseband interface between a baseband unit and a transceiver module of a communication device. The interface enables the bi-directional transmission of digital IQ baseband signals as well as the transmission of control and management information.

The digital interface allows a transmission of IQ samples with rates between 1.2 and 72 MSamples per second. The interface transmits the samples at four different, predefined sample rates between 9.6 and 72 MSamples per second. Thus, waveforms with a bandwidth of up to 57 MHz can be transmitted.

Moreover, the specification supports a division of the IQ samples into up to eight transmitters or receivers in the transceiver module, for example in multiple antenna systems. In this case, the sample rate per transmitter or receiver is reduced, and also the maximum bandwidth for the waveform has been reduced in proportion to the number of transmitters and receivers addressed.

A survey of the transmission modes of the interface is given in Table 6-2 in Chapter 6.3.1.

In the baseband unit, IQ samples transmitted or received are allocated to specific points in time with the assumption of a fixed delay time for the sample rate set by the transceiver module. All other points in time in the transceiver module are referred to in reference to IQ samples.

3 Introduction

Conventional radios systems have been tailored to the use of one dedicated communication standard (waveform). The great variety of communication standards—existing and in planning stage—in the military sector and increasingly in the civilian sector requires the use of several radio sets if different waveforms shall be used.

In the future, so-called Software Defined Radios (SDR) shall more and more overcome this restriction. A significant part of the signal processing of the radio shall be implemented by means of programmable and reconfigurable hardware (e.g. FPGA, DSP). Due to the, in parts, high-standard requirements of level dynamics at the antenna connector of such devices the high-frequency range of these devices is often restricted in its frequency bandwidth. Thus, a modular architecture with a frequency-independent, reconfigurable baseband unit (BBU) and one or more frequency-specific transceiver modules are an appropriate choice (see Figure 3-1).

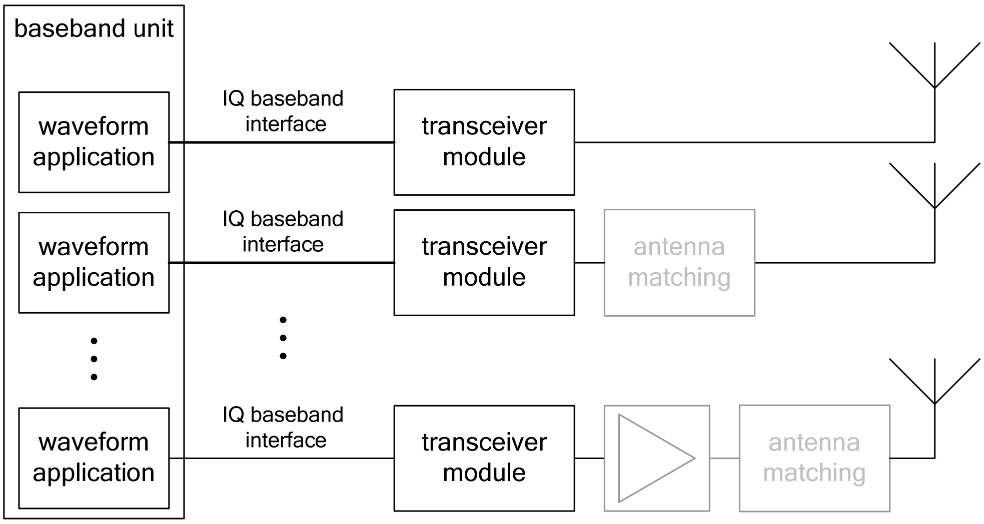


Figure 3-1: Modular SDR Architecture

The interface designated in Figure 3-1 as IQ baseband interface serves to connect various transceiver modules to one waveform application within the BBU. The transceiver module is connected to an antenna with the option to add an external power amplifier or an antenna-matching device between the transceiver module and the antenna.

For mobile radio applications, there are the CPRI and OBSAI initiatives—each of them have defined a standard for the interface between the digital baseband unit and the radio-frequency transceivers. Whereas CPRI has only defined the interface protocols for UMTS and WiMAX, the OBSAI standard has further reaching features (specification of various interfaces, electrical and mechanical specification of the interfaces, GSM and CDMA2000 as additional standards). These two standards provide a basis for the specification of the baseband interface as described in this document. To allow a larger choice of waveforms to be transmitted and a simple and cost-effective implementation, the abovementioned standards are abandoned, when it seems reasonable and technically necessary.

4 Objective

In this document, the interface between a digital baseband unit and a transceiver module is specified.

This specification contains the definition of the transmission protocol and details about timing and synchronization. The specification of the electrical, optical and mechanical parameters of the interface (e.g. voltage levels, wavelengths, connectors, mechanical dimensions) is not subject of this document. The contents of control, test and management information, which are transmitted over the interface along with the IQ data, are not part of this document either.

When preparing this specification, the following aspects have been mainly taken into account:

- fitness for the future
- simple and cost-effective implementation of the interface (both, in the BBU as well as in the transceiver module)
- high flexibility[1]

---

[1] If the objective of additional flexibility conflicts with the simple and cost-effective implementation, the feature of simple implementation has been preferred.

5 Reference Documents

[1] Open Base Station Architecture Initiative (OBSAI) – BTS System Reference Document, Version 2.0

[2] Open Base Station Architecture Initiative (OBSAI) – Reference Point 1 Specification, Version 2.0

[3] Open Base Station Architecture Initiative (OBSAI) – Reference Point 3 Specification, Version 4.0

[4] Common Public Radio Interface (CPRI); Interface Specification, Version 3.0 (2006-10-20)

6 Basic Architecture

The IQ baseband interface specified in this document (see Figure 3-1) is a bi-directional, digital, full-duplex point-to-point connection from a baseband unit to a transceiver module (TRX) with one physical connection for the transmit direction (from the BBU to the transceiver module, here named TX) and another one for the receive direction (from the transceiver module to the BBU, here named RX) are to be provided for each (see Figure 6-1).

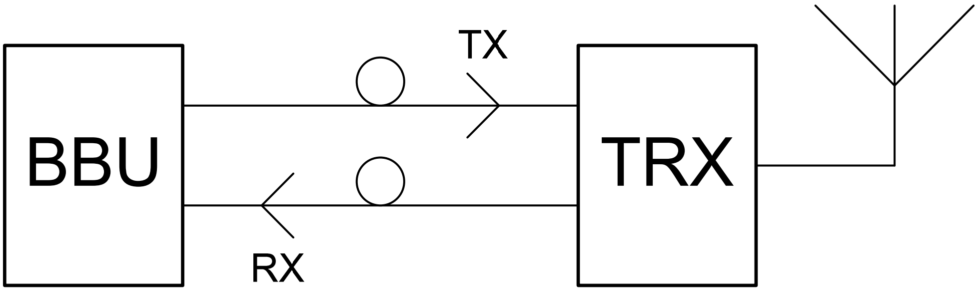


Figure 6-1: IQ Baseband Interface as a Point-to-point Connection between baseband unit BBU and transceiver module TRX

The transmission protocol allows a connection from the baseband unit to several transmitter and/or receiver paths on one transceiver module. Thus, new technologies, such as MIMO and antenna beamforming, can be implemented as well. However, the IQ baseband interface remains a point-to-point connection between baseband unit and transceiver module. The transceiver module distributes data to the individual paths. As an example, Figure 6-2 illustrates the connection of the BBU with a transceiver module 1 with n transceivers as well as a transceiver module m with a transmitter and a diversity receiver.

The baseband unit and the transceiver module are separated in the digital baseband, i.e. digital in-phase (I) and quadrature-phase signals (Q) are transmitted. In addition to the IQ data, information on synchronization as well as on configuration, monitoring and control of the transceiver module are transmitted.

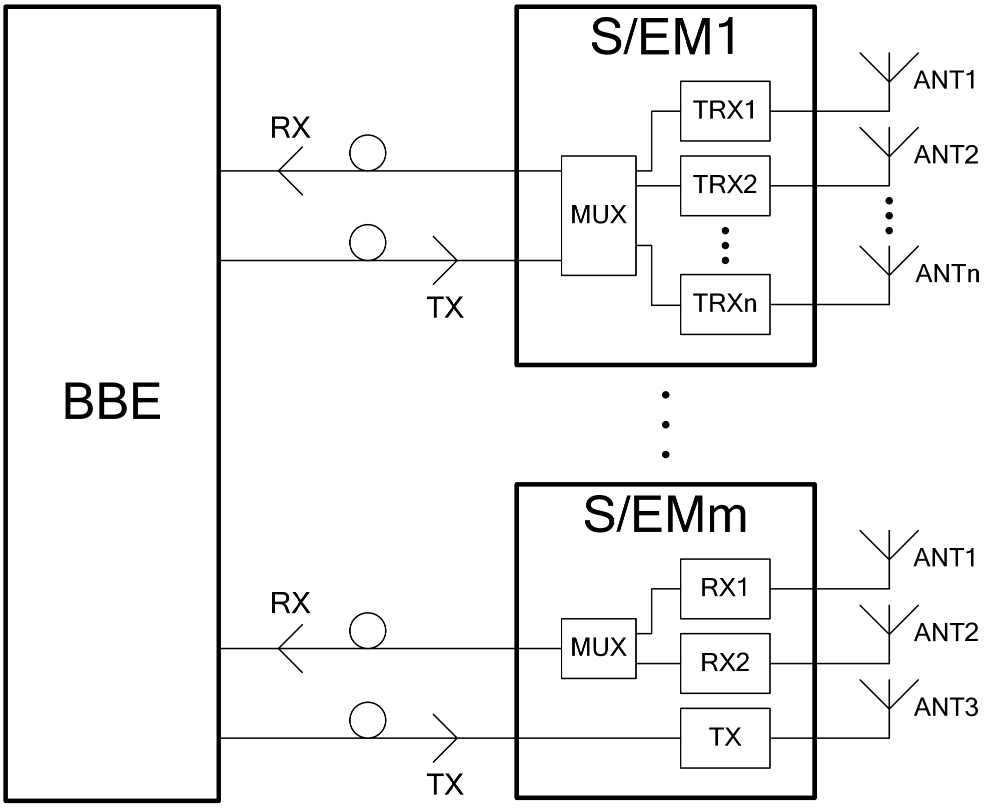


Figure 6-2: Example of the Architecture for Multi antenna Applications (e. g. Beamforming, MIMO)

Figure 6-3 illustrates an example of a possible implementation of the IQ baseband interface. For this purpose, essential components of the interface including blocks of the BBU and of the transceiver module relevant for data transmission are depicted schematically.

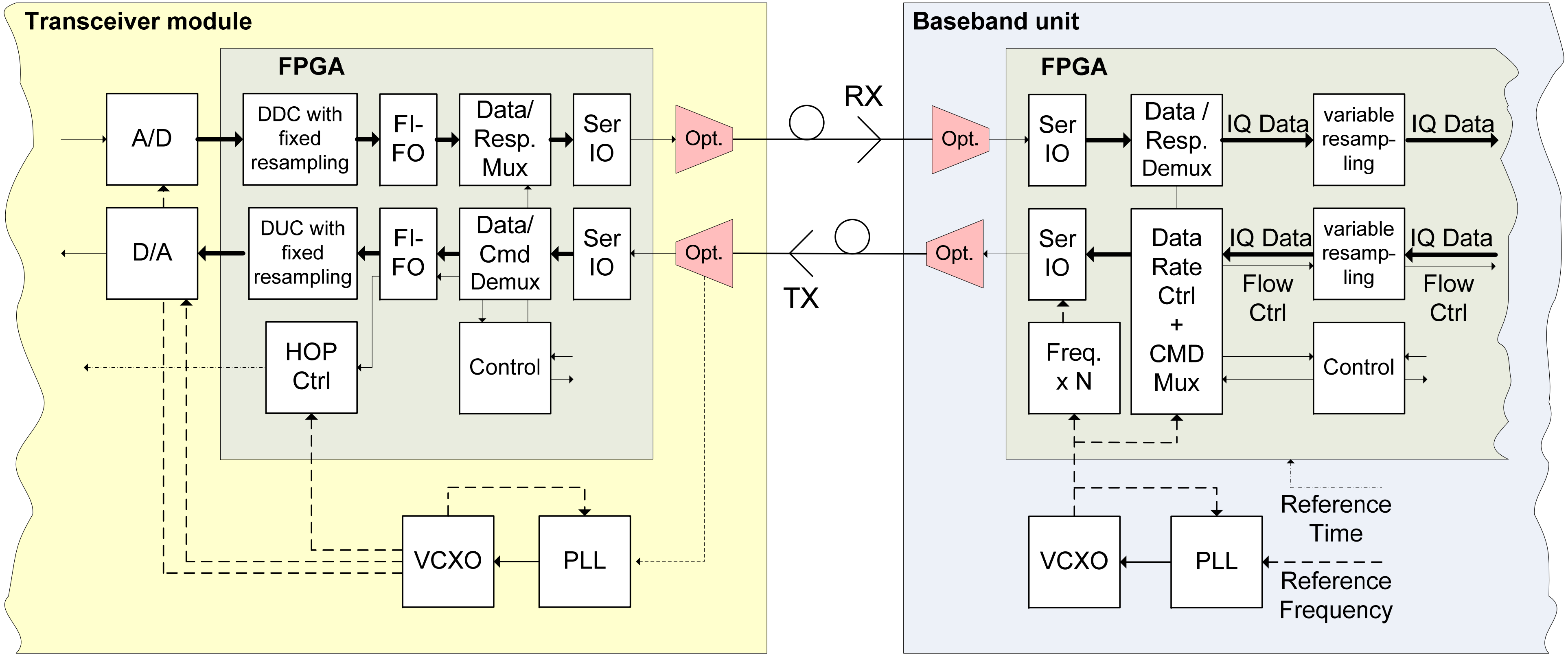


Figure 6-3: Example for the Implementation of the IQ Baseband Interface

6.1 IQ Sample Word Lengths

Three word lengths for IQ samples have been specified:

- In transmit direction (standard)
  I sample and Q sample, 24-bit integer each (in total, 48 bits per IQ sample)
- In receive direction (standard)
  I sample and Q sample, 22-bit mantissa each and 4-bit common exponent (in total, 48 bits per IQ sample)
- In transmit and receive direction (high-rate, optional)
  I sample and Q sample, 16-bit integer each (in total, 32 bits per IQ sample)

6.2 Number of Transmit/Receive Paths on a Transceiver Module

In general, one transmit and one receive path per transceiver module are assumed. For future extensions, such as adaptive beamforming or MIMO, more transmit and/or receive paths can be implemented on a transceiver module (see Figure 6-2). In this case, the number of transmit chains may differ from the number of receive chains.

The number of transmit chains and/or receive chains is limited to eight chains each.

6.3 Transmission Capacity

The IQ baseband interface provides a gross data rate of 768 MBit/s, 1536 MBit/s (=2*768 MBit/s) or 3072 MBit/s (=4*768 MBit/s). Due to the 8B/10B coding scheme used for transmissions, the net data rate is reduced in accordance with Table 6-1.

Factor	i = 1	i = 2	i = 4
Gross data rate	768 MBit/s	1536 MBit/s	3072 MBit/s
Net data rate	**614.4 MBit/s**	**1228.8 MBit/s**	**2457.6 MBit/s**

Table 6-1: Data Rate over the Serial Interface

The net transmission capacity is shared by the *data payload* and the *control payload* (as well as by the *message header* for each *message*). For more details, see Chapter 7.2.

6.3.1 Data Payload

The result of the gross data rate over the IQ baseband interface and the word length defined for an IQ sample is four RF bandwidths that can be transmitted over the interface. When calculating the maximum RF bandwidth of a waveform, it is assumed that a maximum of 80% of the Nyquist bandwidth can be used as RF bandwidth to limit filtering efforts.

Mode	A	B	C	D
Data rate over the interface in **MBit/s**	768	1536	3072	3072
Data format for IQ sample	48 bits[2]	48 bits[2]	48 bits[2]	32 bits[3]
Sampling rate in **MSample/s**	9.6	19.2	48	72
Maximum bandwidth of the waveform in MHz[4]	**7.5**	**15**	**38**	**57**

Table 6-2: Transmission Capacity of the Data Payload

For more IQ sample format details, see Chapter 7.4.3.

6.3.2 Control Payload

The transmission capacity available for the *control payload* depends on the gross data rate of the IQ baseband interface and the transmission capacity defined for the *data payload*.

To be able to exclude transmission errors over the *control payload*, its contents are secured by a cyclic redundancy check (CRC) resulting in a reduced net transmission capacity.

Mode	A	B	C	D
Data rate over the interface in **MBit/s**	768	1536	3072	3072
Gross control payload per *message* (equals 5/3 µs)	26 bytes	58 bytes	26 bytes	26 bytes
Net control payload per *message* (equals 5/3 µs)	24 bytes	56 bytes	24 bytes	24 bytes
Transmission capacity in **MByte/s** (net)	14.4	33.6	14.4	14.4
Transmission capacity in MBit/s (net)	**115.2**	**268.8**	**115.2**	**115.2**

Table 6-3: Transmission Capacity of the Control Payload

For *control payload* content details, see Chapter 7.4.2.

---

[2] The 48 bits are composed of 24-bit integer for an I and/or a Q sample in TX direction each and 22-bit mantissa for an I and/or a Q sample each and 4 bits common mantissa in RX direction

[3] The 32 bits are composed of 16-bit integer for an I and/or a Q sample each

[4] When calculating the maximum bandwidth, it is assumed that only 80% of the Nyquist frequency is used to limit the efforts for the digital filters

7 Transmission Protocol

7.1 Survey

The protocol is defined in such a way that various kinds of information can be exchanged between baseband unit and transceiver module:

- I and Q samples
- synchronization information
- control and management information (CMI); CMI is further divided into
  - time-critical information in close relationship to the IQ samples
  - time-uncritical control and management information

Moreover, the transmission protocol is defined in such a way that it allows for future extensions.

The user data from the different waveforms are transmitted over the serial connection in the form of IQ data. If the BBU is connected to several transmit or receive paths on one transceiver module, the IQ data are transmitted by means of time-division multiplexing.

CMI is transmitted in such a way that time-critical data (such as frequencies in the frequency hopping procedure) are transmitted with a higher priority than time-uncritical data (such as SBIT, OBIT and IBIT). Such information is transmitted in time-division multiplex with IQ data.

7.2 Physical Layer

In this document only parts of the physical layer are described. The electrical, optical and mechanical parameters strongly depend on applications and/or implementations and are not specified in this document.

7.2.1 Data Rate over the Device Interface

For maximum flexibility as well as cost and energy efficiency, various data rates of the IQ baseband interface have been defined:

Data rate option i = 1: 768 MBit/s (Mode A)

Data rate option i = 2: 1536 MBit/s = 2 * 768 MBit/s (Mode B)

Data rate option i = 4: 3072 MBit/s = 4 * 768 MBit/s (Modes C and D)

The BBU must provide all three data rate options, a transceiver module must be able to communicate with a minimum of one option. The data rate from BBU to transceiver module and vice versa is always identical.

7.2.2 Coding of the Serial Data Connection

For the serial data transmission an 8B/10B transmission coding in accordance with the 802.3-2002 IEEE standard shall be used. This ensures a DC-balanced data stream, thus enabling a bit recovery and a clock recovery in the receiver of the serial data stream.

In addition, errors in the connections of the data stream can be detected by the 8B/10B coding, thus enabling a feedback on the connection quality.

For 8B/10B coding details, see [5] and [6].

7.2.3 Bit Error Detection and Correction

No forward error correction (FEC) scheme is provided for the serial data transmission. However, control information (*control payload*) is protected by an inherent cyclic redundancy check (CRC) in order to detect bit errors and, thus, protect the device from malfunction by control information recognized erroneously.

If a CRC error occurs in a transmission from BBU to transceiver module, the *CRC error flag* (CEF) will be set in the *message header* of the subsequent *message* to the BBU (= *message* in RX direction with the same value in the *message counter* as the faulty *message* in TX direction).

Signaling by the CEF shall enable an instant repetition of control packages of the last control payload. The CEF will be re-deleted in the next *message* (also see Chapter 7.4.1). Since the CEF transmission may be disturbed as well, an additional signaling of a CRC error by the transceiver module, e.g. in the form of an alert or status report, is an appropriate choice.

7.3 Data Link Layer

7.3.1 Frame Structure

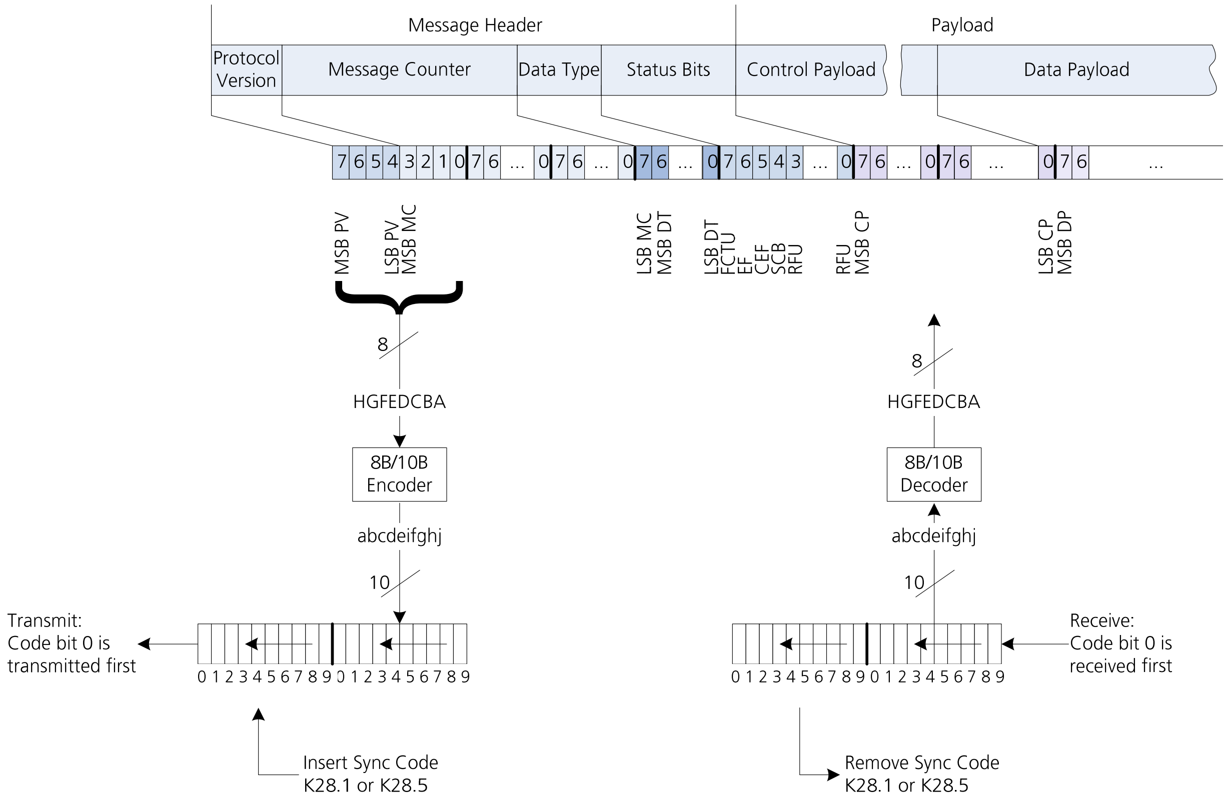


Figure 7-1: Data Structure of a Message

The digital data stream over the serial interface is structured into so-called *messages*. Figure 7-1 illustrates the data structure of a message with 8B/10B coding and the insertion of the *sync code* (K28.1 or K28.5, [5]).

Each *message* is composed of a *message header*, providing information on the content of the *message*, and the *payload*, which is, in turn, composed of a *control payload* and a *data payload*. See Chapters 7.4.1, 7.4.2 and 7.4.3 for further details on *message header* and *payload*.

When transmitted, a *message* is split into single bytes. Transmission is commensurate with the illustration in Figure 7-1 from left to right, i.e. following the *sync code* the *protocol version* and *message counter* of the *message header* are transmitted—just prior to the last byte of the *data payload*. Prior to transmission, each byte is encoded by means of the 8B/10B code. Then, bit "0" is transmitted first.

A *message* has a fixed time length of 5/3 µs (≈ 1.67 µs). Thus, it contains—depending on the option for data rate i (see Chapter 7.2.1)—a number of 128, 256 or 512 bytes. The structure of a *message* for the various serial data rates is shown in Table 7-1.

Field	i = 1	i = 2	i = 4
Message header	6 bytes	6 bytes	6 bytes
Control payload	26 bytes	58 bytes	26 bytes
Data payload	96 bytes	192 bytes	480 bytes
Total length	**128 bytes**	**256 bytes**	**512 bytes**

Table 7-1: Structure of a Message Depending on the Data Rate over the Serial Interface

As illustrated in Figure 7-2, *RX messages* are transmitted with an offset of D_{TRM} to the *TX messages*. D_{TRM} is equal to the length of one *message*.

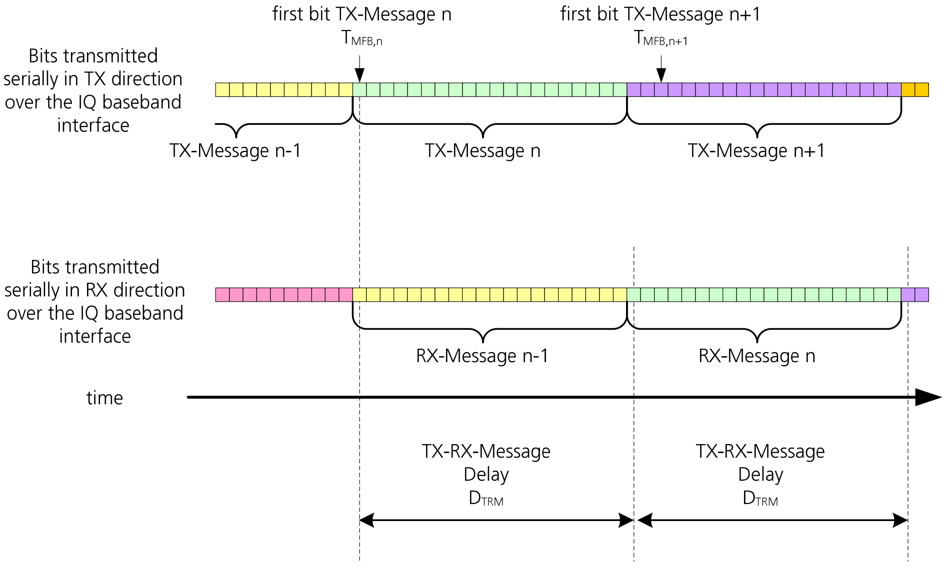


Figure 7-2: Frame Offset of TX/RX Messages

The *RX message* following the *TX message* with number n in a time offset of D_{TRM} receives the same number n in the *message counter*.

7.3.2 Frame Synchronization

Before a transmission over the IQ baseband interface is possible, the transceiver module must be synchronized with the BBU and vice versa.

The frame synchronization processes for both transmitter and receiver are illustrated in the state diagrams in Figure 7-3 and Figure 7-4. The parameters used are shown in Table 7-2:

Parameter	Description
TX_EN	Boolean - is set to enable transmission over the interface
LOS_EN	Boolean - indicates whether the "Loss of Signal" signal of the receiver has an impact on the transmitter state
LOS	Boolean - "Loss of Signal" in the receiver, is set when the receiver is in UNSYNC state
LOF_EN	Boolean - indicates whether the "Loss of Frame" signal of the receiver has an impact on the transmitter state
LOF	Boolean - "Loss of Frame" in the receiver, is set when the receiver changes from SYNC state into WAIT_FOR_K28.1_SYNC state
SYNC_T	Threshold value for consecutive valid blocks in order to change into WAIT_FOR_K28.1_SYNC state Unsigned integer, 16 bits, zero is not permitted Default value = 255
UNSYNC_T	Threshold value for consecutive invalid blocks in order to change into UNSYNC state Unsigned integer, 16 bits, zero is not permitted Default value = 255
BLOCK_SIZE	Defines the number of bytes in one block Unsigned integer, 16 bits, zero is not permitted Default value = 400
SYNC_M	Defines after how many bytes a sync code must be received in order to change into SYNC state SYNC_M = (128 * i) – 1 with i being the data rate option in accordance with Chapter 7.2.1 Unsigned integer, 16 bits, zero is not permitted

Table 7-2: Parameters for Frame Synchronization

Transmitter:

There are three states in the state diagram of the transmitter: OFF, IDLE and TX_BUSY. After re-setting, the transmitter is in OFF state. In this state, no data are transmitted, the transmitter is disabled (electrical: TX inhibit, optical: transmitter does not transmit light).

In the BBU, the application layer controls the transition from OFF state into IDLE state. For this purpose, the TX_EN parameter must be set to "1" and, in addition, one of the following two conditions must be met: (1) LOS_EN is set to "0", i.e. that the LOS (Loss of Signal) signal from the receiver does not have any impact on the transmitter state or (2) LOS_EN is "1" (i.e. LOS has an impact on the transmitter) and LOS is "0" (i.e. no Loss of Signal from the receiver).

LOS_EN must be set to "0" in the BBU in order to be able to start frame synchronization since the transceiver module can only transmit when receiving data from the BBU.

In the transceiver module, the transition from OFF state into IDLE state is controlled by the receiver: The TX_EN parameter is then set to "1" when the receiver changes into WAIT_FOR_K28.1_SYNC state (i.e. valid K28.5 IDLES are received by the BBU).

In IDLE state, K28.5 IDLE bytes are transmitted continuously. This enables the receiver to synchronize the PLL of its reference clock to the bit clock of the interface as well as the interface with the individual samples (bytes).

The state machine in the BBU changes from IDLE state into TX_BUSY state when the receiver changes into WAIT_FOR_K28.1_SYNC state (i.e. valid K28.5 IDLES are received). In the transceiver module, the transition into TX_BUSY state takes place when the receiver changes into SYNC state (i.e. valid *messages* are received).

In the TX_BUSY state, valid *messages* are transmitted over the interface. The transceiver module transmits its *messages* to the BBU, offset by D_{TRM} (see Chapter 7.3.1).

The transmitter changes from TX_BUSY state into IDLE state when the receiver state machine sets the LOF parameter to "1" and LOF_EN has been set to "1" at the same time. In the BBU, it may be an appropriate choice to ignore LOF (i.e. LOF_EN = "0") when the frame synchronization of BBU and transceiver module is monitored in a higher layer.

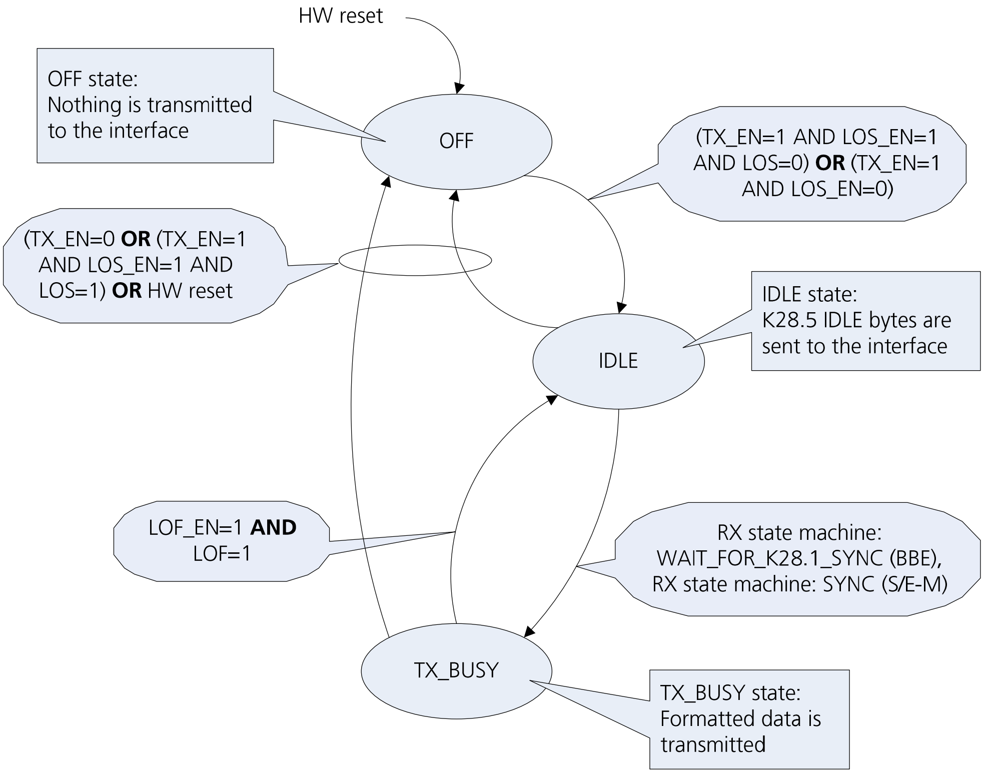


Figure 7-3: Transmitter State Diagram

The transmitter changes into OFF state again if (1) a HW reset is initiated, (2) LOS is triggered in the receiver (in case LOS_EN=1) or (3) the transmitter is disabled by the application layer or by loss of synchronization within the receiver (TX_EN=0).

Receiver:

There are four states in the state diagram of the receiver: UNSYNC, WAIT_FOR_K28.1_SYNC, WAIT_FOR_MESSAGE and SYNC.

Two criteria are applied to synchronize the receiver: (1) the byte error rate is determined and (2) the reception of valid *message* lengths is monitored.

After re-setting, the receiver is in UNSYNC state. In this state, either no bytes or a great number of invalid bytes are received. The transition into WAIT_FOR_K28.1_SYNC state takes place if SYNC_T correct, consecutive blocks of bytes were received. A block is considered to be received correctly if there are no errors in the block during the 8B/10B decoding process. The SYNC_T and BLOCK_SIZE parameters must be defined in advance.

As soon as a valid K28.1 *sync code* is received, the receiver changes into WAIT_FOR_MESSAGE state. In this state, the receiver waits for the next *sync code* (either K28.1 or K28.5) while the number of bytes received between the *sync codes* is analyzed. If the next *sync code* is received after SYNC_M bytes, the receiver changes into SYNC state and a regular data transmission may commence—otherwise the receiver returns into WAIT_FOR_K28.1_SYNC state.

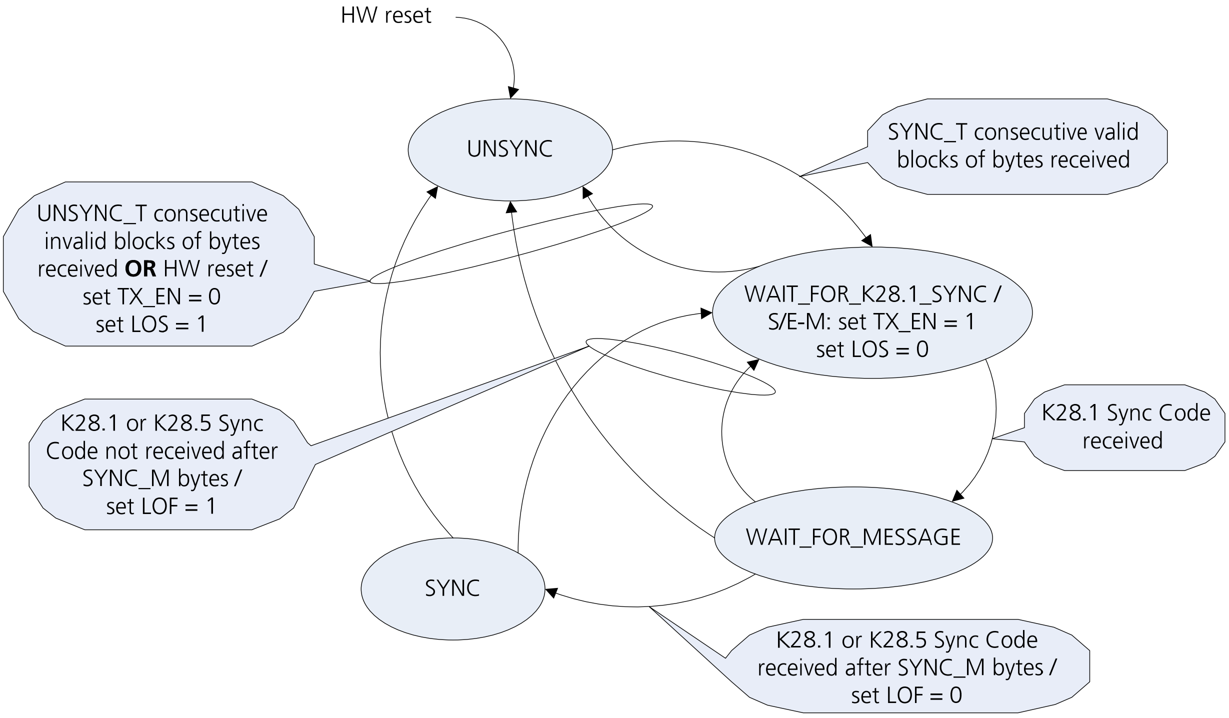


Figure 7-4: Receiver State Diagram

The SYNC_M parameter value is commensurate with the data rate over the interface and can be calculated:

SYNC_M = (128 * i) – 1, with i being the data rate in accordance with Chapter 7.2.1.

Only in SYNC state, the link layer transfers *messages* to the application layer.

If, in SYNC state, a K28.1 or K28.5 *sync code* is too late or too early so that between both *sync codes* not SYNC_M bytes have been received, the receiver changes into WAIT_FOR_K28.1_SYNC state. At the same time, this signals the loss of frame synchronicity to the transmitter's state machine by setting LOF = "1".

After HW reset or if a minimum of UNSYNC_T consecutive invalid blocks are received, the receiver changes from any state into UNSYNC state immediately. A block is invalid if a minimum of one of its bytes has an error during the 8B/10B decoding process.

When the receiver changes from SYNC state into UNSYNC state TX_EN is set to "0". The BBU only transmits again when TX_EN is set to "1" by the application layer. This ensures a new frame synchronization after a major transmission error over the interface.

7.3.3 Automatic Negotiation of the Data Rate over the Interface

Transmission over the IQ baseband interface features different data rates (see Chapter 7.2.1). The BBU must be able to communicate with all data rates, a transceiver module must be able to communicate with a minimum of one data rate.

An automatic data rate negotiation shall be performed if the data rate had not been defined in advance (e.g. when the data rate on which the transceiver module can communicate is already known to the BBU).

Table 7-3 lists the parameters which are used for negotiating the data rate. MaxTxTime must have a sufficient length to enable a successful synchronization. MaxRxTime should have a minimum length of three times MaxTxTime in order to be able to check all three supported data rates of the BBU with each data rate of the transceiver module.

MaxTxTime, MaxRxTime as well as MaxSynchronizationTime (as time limit after which the automatic data rate negotiation will be stopped) are indicated as a multiple of the word clock (16 bits each; these are equal to 20 bits over the interface due to 8B/10B coding) over the interface with i being commensurate with the data rate option in accordance with Chapter 7.2.1.

Parameter	**Description**
Synchronization	Boolean – indicates whether synchronization and data rate negotiation in the BBU are completed successfully or not
RxSynchronization	Boolean – indicates whether synchronization and data rate negotiation in the transceiver module are completed successfully or not
TimeOutCounter	Counter – if it reaches a defined value (*MaxSynchronizationTime*), the data rate negotiation is stopped
MaxTxTime	Maximum transmission time at a specific data rate This value is indicated as a multiple of the word clock. Default value = i * 7,680,000 (200 ms)
MaxRxTime	Maximum receipt time at a specific data rate This value is indicated as a multiple of the word clock. Default value = i * 24,576,000 (0.64 s)
MaxSynchronizationTime	Time limit for the data rate negotiation This value is indicated as a multiple of the word clock. Default value = i * 192,000,000 (5 s)

Table 7-3: Parameters for the Data Rate Negotiation

Algorithm for the Baseband Unit:

1. Set *Synchronization*=FALSE and start a time-out counter (*TimeOutCounter*).
2. Select the lowest data rate supported by the BBU (i=1).
3. Attempt synchronization with the transceiver module by performing steps 3a–3c. After a maximum period of *MaxTxTime*, go to step 4.
    a) Transmit K28.5 IDLE bytes to the state machine for the frame synchronization of the receiver in the transceiver module.
    b) When the receiver state machine of the BBU enters WAIT_FOR_K28.1_SYNC state since it has received valid K28.5 IDLES from the transceiver module, start a transmission to the transceiver module in the valid *message* format.
    c) When the state machine enters SYNC state for the frame synchronization of the BBU receiver, set *synchronization*=TRUE (i.e. the synchronization and the data rate negotiation have been completed).
4. If *Synchronization*=FALSE and *TimeOutCounter* is less than *MaxSynchronizationTime*, change to the next higher data rate supported or to the lowest data rate if the highest BBU data rate had already been reached. Return to step 3.
5. End of algorithm.

Algorithm for the Transceiver Module:

1. Set *RxSynchronization*=FALSE and start a time-out counter (*TimeOutCounter*).
2. Select the lowest data rate supported by the transceiver module.
3. Attempt synchronization with the BBU by performing steps 3a–3c. After a maximum period of *MaxRxTime*, go to step 4.
    a) Start the receiver state machine for frame synchronization (see Chapter 7.3.2).
    b) When the receiver state machine enters WAIT_FOR_K28.1_SYNC state since it has received valid K28.5 IDLES from the BBU, start a transmission of K28.5 IDLE bytes back to the BBU.
    c) When the receiver state machine enters SYNC state, start a transmission to the BBU in the valid *message* format and set *RxSynchronization*=TRUE.
4. If *RXSynchronization*=FALSE and *TimeOutCounter* is less than *MaxSynchronizationTime*, change to the next higher data rate supported or to the lowest data rate if the highest data rate supported by the transceiver had already been reached. Return to step 3.
5. End of algorithm.

7.3.4 Commanded Change of the Data Rate over the Interface

If the BBU intends to start a waveform that requires a higher data rate over the interface as set currently, a change of the data rate must be commanded to the transceiver module. Since synchronization has already been conducted and data in the *message* format are exchanged, the transceiver module data rates available can be retrieved.

If the transceiver module can communicate with the higher data rate required, the transceiver unit is commanded with a *TU command* in the *control payload* to set a new data rate (see Chapter 7.4.2.4.4).

Table 7-4 lists the parameters used for changing the data rate. MaxTxTime must have a sufficient length to enable a successful synchronization.

Parameter	**Description**
Synchronization	Boolean – indicates whether synchronization and data rate negotiation in the BBU have been successfully completed or not
RxSynchronization	Boolean – indicates whether synchronization and data rate negotiation in the transceiver module have been successfully completed or not
NewDataRate	Boolean – signals whether the new data rate has been set or not
MaxTxTime	Maximum transmission time at a specific data rate

Table 7-4: Parameters for Commanded Data Rate Negotiation

Algorithm for the Base Band Unit:

1. Set *Synchronization*=FALSE.
2. Select the intended data rate to be changed to and set *NewDataRate*=TRUE.
3. Attempt synchronization with the transceiver module by performing steps 3a–3c. After a maximum period of *MaxTxTime*, go to step 4.
    a) Transmit K28.5 IDLE bytes to the state machine for the frame synchronization of the receiver in the transceiver module.
    b) When the receiver state machine of the BBU enters WAIT_FOR_K28.1_SYNC state since it has received valid K28.5 IDLES from the transceiver module, start a transmission to the transceiver module in the valid *message* format.
    c) When the state machine of the BBU enters SYNC state for frame synchronization, set *Synchronization*=TRUE (i.e. synchronization and data rate negotiation have been completed).
4. If *Synchronization*=FALSE and *NewDataRate*=TRUE, change to the data rate that was first set and had allowed synchronization. Set *NewDataRate*=FALSE and return to step 3.
5. End of algorithm.

Algorithm for the Transceiver Module:

1. Set *RxSynchronization*=FALSE.
2. Select the data rate intended and set *NewDataRate*=TRUE.
3. Attempt synchronization with the BBU by performing steps 3a–3c. After a maximum period of *MaxTxTime*, go to step 4.
    a) Start the receiver state machine for frame synchronization (see Chapter 7.3.2).
    b) When the receiver state machine enters WAIT_FOR_K28.1_SYNC state since it has received valid K28.5 IDLES from the BBU, start a transmission of K28.5 IDLE bytes back to the BBU.
    c) When the receiver state machine enters SYNC state, start a transmission to the BBU in the valid *message* format and set *RxSynchronization*=TRUE.
4. If *RXSynchronization*=FALSE and *NewDataRate*=TRUE, change to the data rate that was first set and had allowed synchronization. Set *NewDataRate*=FALSE and return to step 3.
5. End of algorithm.

7.4 Transport Layer

7.4.1 Message Header

The *message header* has a fixed length of 6 bytes. Size, contents and function of the individual *message header* components are illustrated in Figure 7-5 and Table 7-5.

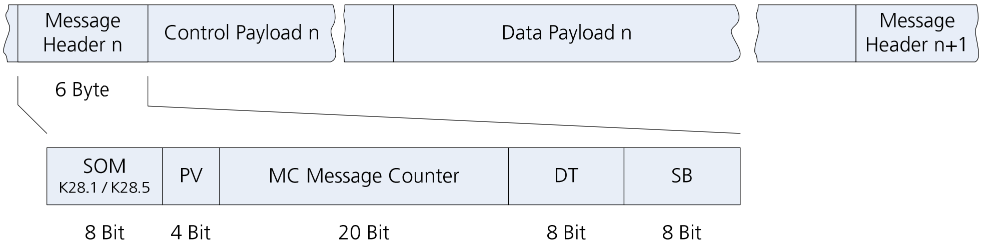


Figure 7-5: Definition of the Message Header

Field	Description
SOM *Length*	*Start Of Message* *(8 bits)* At the beginning of each *message* a so-called *sync code* is inserted with the K28.1 and K28.5 *sync codes* (*comma*) being used on an alternating basis (see [5]). The K28.1 *sync code* is used for *messages* with an even-numbered *message counter* (incl. 0). For *messages* with an odd-numbered *message counter* the K28.5 *sync code* is used. A *sync code* is a 10-bit word with no 8-bit value correlation in the 10B/8B decoding and that, hence, can be positively identified in the data stream. It marks the *message* limits. In addition, the K28.5 *sync code* is used as so-called *comma* for byte synchronization. The *sync code* can be inserted after 8B/10B coding only. When calculating the length of the *message header* and/or the *message*, it is taken into account as one byte. See [5] for details about 8B/10B coding and K-Codes.
PV *Length*	*Protocol Version* *4 bits* The *protocol version* indicates the number of the version of this "Specification of the IQ Baseband Interface" document the software implementation is based upon. The first bit of the *protocol version* is always "0" (Reserved for Future Use - RFU) => PV = 0b0xxx. This version of the document is referred to in the *message header* by 0b0001.
MC *Length*	*Message Counter* *20 bits* The message counter counts the *messages* from 0 to 599 999, i.e. exactly one second and re-starts with "0" after a reset.
DT *Length*	*Data Type* *8 bits* On the one hand, the *data payload* format is specified and, on the other hand, it is signaled whether valid IQ samples are transmitted within the *message*. Bit 0 is set if valid IQ samples are transmitted. Data are invalid, for example, if the transmitter or the receiver is disabled and the interface, however, is active. Bit 1 specifies the length of an IQ sample. Bit 1 is set if 4 bytes

Field	Description
	per IQ sample are used only (mode D). Bits 2 and 3 have not yet been defined (shall be transmitted with "0" in the transmitter and be ignored in the receiver). Bits 4 to 7 indicate how many transmitters or receivers per transceiver module are addressed. For this purpose, only one of the four bits can be set each. Bits 4 to 7 can also be set to reduce the sample rate with one transmitter or receiver on the transceiver module (for example, setting bit 5 means reducing the sample rate by half, setting bit 6 means reducing it to ¼). At the moment, the following types have been defined: 0x00 = data undefined, e.g. use in init phase 0x10 = *data payload* with 6 bytes per IQ sample, 1 transmitter / receiver, data invalid 0x11 = *data payload* with 6 bytes per IQ sample, 1 transmitter / receiver, data valid 0x12 = *data payload* with 4 bytes per IQ sample, 1 transmitter / receiver, data invalid 0x13 = *data payload* with 4 bytes per IQ sample, 1 transmitter / receiver, data valid 0x20 = *data payload* with 6 bytes per IQ sample, 2 transmitters / receivers, data invalid 0x21 = *data payload* with 6 bytes per IQ sample, 2 transmitters / receivers, data valid 0x22 = *data payload* with 4 bytes per IQ sample, 2 transmitters / receivers, data invalid 0x23 = *data payload* with 4 bytes per IQ sample, 2 transmitters / receivers, data valid 0x40 = *data payload* with 6 bytes per IQ sample, 4 transmitters / receivers, data invalid 0x41 = *data payload* with 6 bytes per IQ sample, 4 transmitters / receivers, data valid 0x42 = *data payload* with 4 bytes per IQ sample, 4 transmitters / receivers, data invalid 0x43 = *data payload* with 4 bytes per IQ sample, 4 transmitters / receivers, data valid 0x80 = *data payload* with 6 bytes per IQ sample, 8 transmitters / receivers, data invalid 0x81 = *data payload* with 6 bytes per IQ sample, 8 transmitters / receivers, data valid 0x82 = *data payload* with 4 bytes per IQ sample, 8 transmitters / receivers, data invalid 0x83 = *data payload* with 4 bytes per IQ sample, 8 transmitters / receivers, data valid

Field	Description
SB	*Status Bits*
Length	*8 bits*
	1st bit (MSB): *Flow Control Time-Uncritical Control Data* - FCTU 0b1xxxxxxx = the receiver can still accept a minimum of 3 time-uncritical *control packets*. Upon transition from "1" to "0", the receiver can still accept a minimum of 2 time-uncritical *control packets*. The FCTU is used for the signaling from the transceiver module to the BBU only, in counter-direction its value is always "1". 2nd bit: *Event Flag* - EF The *event flag* serves to report to the BBU a change of state in the transceiver module. Hence, it can, for example, signal an error or a critical state. The EF continues to be set for as long as the BBU transmits a command to delete the EF after retrieval and successful receipt of the status. If another event not yet retrieved exists, the EF continues to be set until this event has also been transmitted successfully. An event transmitted may only be deleted in the transceiver module if the command to delete the EF has been received. If an event is retrieved before the command to delete the EF has been received by the transceiver module, the event will be re-transmitted. The EF is used for signaling from the transceiver module to BBU only, in counter-direction its value is always "0". 0bx1xxxxxx = a status report (e.g. alert) is available to be collected by the receiver. 3rd bit: *CRC Error Flag* – CEF The *CRC error flag* is set in case a CRC error occurs when receiving the last *control payload*. It is reset with the next message unless another CRC error has occurred. The CEF is used for signaling from the transceiver module to BBU only, in counter-direction its value is always "0". 0bxx1xxxxx = in the *control payload* of the last *message* received a CRC error occurred. 4th bit: *Serial Communication Bit* – SCB The *serial communication bit* is available for a serial communication path outside the protocol described herein. Shall be transmitted in the transmitter with value "0". Shall be ignored in the receiver. 5th - 8th bit: *Reserved for Future Use* – RFU

Field	Description
	Shall be transmitted in the transmitter with value "0". Shall be ignored in the receiver.

Table 7-5: Definition of the Message Header

7.4.2 Control Payload

7.4.2.1 Control Payload Format

In the *control payload*, control and management information is transmitted from the baseband unit to the transceiver modules and status reports are transmitted from the transceiver module to the baseband unit.

In general, there are three types of information:

- time-critical information which is correlated to IQ samples received; therefore, it must be transmitted with real-time requirements
- time-uncritical information; for the transmission of this type of information no real-time requirements apply
- padding characters (*padding*)

Figure 7-6 illustrates the structure of the *control payload*.

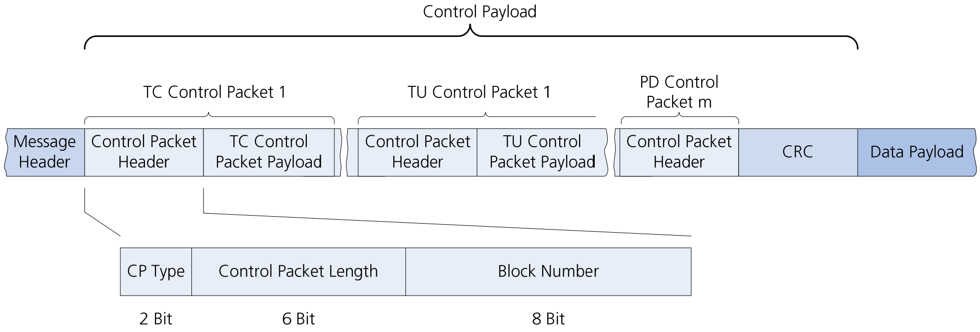


Figure 7-6: Control Payload Structure

Within the *control payload* of a *message* several time-critical (TC), several time-uncritical (TU) and padding (PD) data packets (*control packets*) can be transmitted. The length of the *control payload* is fixed for a data rate defined. *Padding* is used to fill the *control payload* up to the length defined.

7.4.2.2 Control Packet

The structure of a *control packet* is illustrated in Figure 7-7.

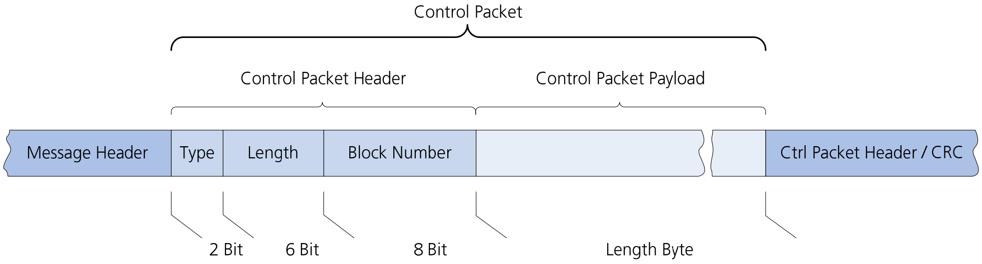


Figure 7-7: Control Packet Structure

A *control packet* is composed of a *control packet header* and a *control packet payload*. The structure of *control packets* is uniform for time-critical and time-uncritical data as well as *padding*.

The *control packet header* has a fixed length of two bytes and is identical for time-critical and time-uncritical data packets as well as *padding*. Size, contents and function of the individual components of the *control packet header* are illustrated in Table 7-6 and Figure 7-6.

Field	Description
Control Packet Type *Length*	Type of the data packet (*control packet*) *2 bits, including MSB* 10 = TC – time-critical information 01 = TU – time-uncritical information 11 = PD – *padding* (fill data)
Control Packet Length *Length*	Length of the *control packet payload* *6 bits – including LSB* This field indicates the number of bytes used (i.e. with no fill byte) within the *control packet payload*. In case of *padding*, the *control packet payload* and, hence, the *control packet length* is always zero.
Block Number *Length*	The *block number* numbers the individual *control blocks* of a *command*. *8 bits* The higher-value bit indicates whether it is the last *control block* within a *command* or not. The *control blocks* of the *command* are enumerated by means of the lower 7 bits. 0b1xxxxxxx = last *control block* of the *command* 0b00000000 = first *control block* of the *command* 0b00000001 = second *control block* of the *command* … 0b10001001 = tenth and last *control block* of the *command* A *TC command* and *padding* are always transmitted completely in one *control packet*. Thus, the *block number* here is always 0b10000000.

Table 7-6: Definition of the Control Packet Header

The *control packet payload* contains a *TC command* or a *TU command* and/or a *TU control block* as part of a *TU command*. For faster processing by means of 16-bit read or write access a *fill byte* (0x00) is attached to a *control packet* with an uneven number of bytes. By means of the *control packet length* this byte can be identified in the receiver and be removed.

In case of *padding*, the length of the *control packet payload* is equal to "0", i.e. one *control packet header* only is transmitted.

7.4.2.3 Time-critical Information (TC)

If time-critical information is to be transmitted, it will be transmitted at the beginning of the *control payload* within the *control packet payload* of a *TC control packet*. Time-critical data packets cannot be distributed to several *messages*.

Figure 14-1 illustrates the way a *TC command* is assembled into a *control packet* and inserted into the *control payload* of a *message*.

7.4.2.3.1 TC Control Packet Payload Format

The *TC control packet payload* is the transport medium for a *TC command*. The length of a *TC control packet payload* depends on the length of a *TC command* to be transmitted.

A *TC command* and, hence, the *TC control packet payload* must not exceed a length of 22 bytes (26 bytes minimum length of the *control payload* minus 2 bytes of the *control packet header* and 2 bytes of the CRC) so that it can be transmitted within the *control payload* of a single *message* (this maximum length for a *TC command* also applies to mode B although the *control payload* has a length of 58 bytes in this mode).

If a CRC error is determined in the transmission of a *control payload* containing a *TC control packet*, this *TC control packet* and, hence, the *TC command* are discarded (see 7.4.2.6).

7.4.2.3.2 TC Command Format

Figure 7-8 illustrates the structure of a *TC command*.

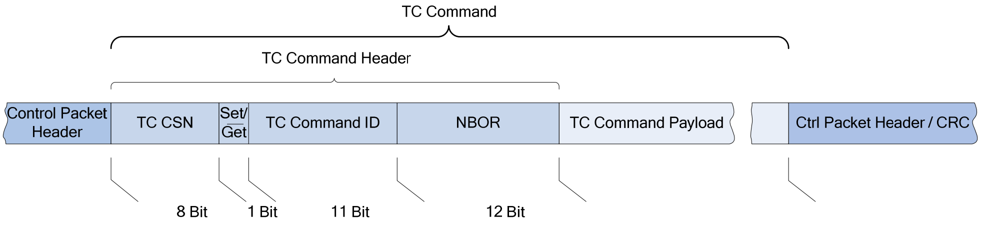


Figure 7-8: TC Command Format

The components of the *TC command header* are described in Table 7-7.

Field	Description
TC CSN *Length*	The *TC command sequence number* numbers the individual *TC commands*. *8 bits* If the counter reaches 0xFF, it is set to 0x00 with the next *TC command*. In this way, a number of 256 *TC commands* can be clearly referenced.
Set/$\overline{\text{Get}}$	By means of the Set/$\overline{\text{Get}}$ bit it is signaled whether a value shall be set or read.

Length	*1 bit* 0 = read 1 = set
TC Command ID *Length*	The *TC command ID* defines which command is transmitted. *11 bits*
NBOR *Length*	By means of the NBOR parameter the action delay is transmitted as number of samples (N_{BOR} - *BBE Offset Reference* = the offset to the reference point which is specified by the BBU, see Chapter 8.1.4.1). *12 bits* The range of values from 0x000 to 0xFFE is used to delay the action by 0 to 4094 samples (unsigned integer). NBOR is set to the maximum value of 0xFFF if an action shall be performed immediately, i.e. without any association with a sample. 0xFFF = action is performed immediately 0x0C6 = action is delayed by 198 samples For further details, see Chapter 8.1.4.

Table 7-7: TC Command Header Definition

Within the *TC command payload* the data required are transmitted commensurate with the *TC command ID*.

Each *TC command* is associated with a *TC command sequence number* transmitted within the *TC command header*. This number enables detection of lacking *TC control packets* by means of a transmission error in the receiver.

7.4.2.3.3 TC Application Response

No acknowledge to a *TC command* is transmitted on transport level. However, it receives a *TC application response* if it is a Get command. A time-critical Set command receives neither a transport acknowledge nor a *TC application response*.

In case of an *application response*, the TC CSN, the Set/$\overline{\text{Get}}$ bit and the *TC command ID* of the *TC command* are inherited in the *TC command header*. The TC CSN in conjunction with the *TC command ID* is used to enable an association of the *TC application response* with the pertinent *TC command*.

The components of the *TC application response header* are described in Table 7-8.

Field	Description
TC CSN	*TC command sequence number* of the *TU command* for which a response is expected.
Length	*8 bits*
Set/$\overline{\text{Get}}$	The Set/$\overline{\text{Get}}$ bit is inherited from the pertinent *TC command* immediately and indicates that this *TC command* has been a read access (Get).
Length	*1 bit* 0 = Get 1 = Set
TC Command ID	Here, the *TC command ID* of the *TC command* for which a response is expected is transmitted.
Length	*11 bits*
NBOR	By means of the NBOR parameter the action delay is transmitted as number of samples (N_{BOR} - *BBE Offset Reference* = the offset to the reference point which is specified by the BBU, see Chapter 8.1.4.1).
Length	*12 bits* This parameter has no meaning for an *application response* and is transmitted with "0".

Table 7-8: TC Application Response Header Definition

Within the payload of the *TC application response* the parameter retrieved is transmitted.

7.4.2.4 Time-uncritical Information (TU)

Time-uncritical information can only be transmitted within the *control payload* of a *message* when the *control payload* has not been completely filled up by time-critical information packets. Time-uncritical information can have various lengths from a few bytes (e.g. writing or reading a parameter) to several kilobytes (e.g. SW or HW update) and must be distributed to a number of *messages*, if required. In this way, larger blocks of data, such as file transfers, can be transmitted.

Figure 14-2 in the Annex illustrates the way a larger file is split into individual *TU commands*, packed into individual *TU control packets* and inserted into the *control payload* of the *messages*.

7.4.2.4.1 TU Control Packet Payload Format

The *TU control packet payload* is the transport medium for a *TU command*. The length of a *TU control packet payload* depends on the length of the *TU command* to be transmitted.

An interruption in the transmission of a *TU command* by a *TU command* of higher priority is not intended. A large file is split into so-called *partitions* with a maximum size of 1020 bytes by the application layer so that the file's transmission does not block the transmission channel. The *partition* is supplemented by a 4-byte header to a *TU command* with a maximum of 1024 bytes.

If a *partition* has an exact length of 1020 bytes, this indicates that a minimum of one more *partition* of this data packet will follow. If a file has been transmitted completely and its last partition had 1020 bytes exactly, one more *TU command* with a *TU command length* of "0" (i.e. with no *TU command payload*) must be transmitted to signal the end of the file.

A *next partition* bit in the *TU command header* signals that the *partition* in the payload of the *TU command* must be attached to a previously transmitted *partition* of a larger file (> 1020 bytes). Individual *partitions* of a file are assembled in the chronological order of their receipt in the receiver so they must be split into *partitions* and transmitted in the correct order.

If the *next partition* bit has not been set (NP = "0") although the file has not been completely transmitted, the file transfer will be re-started with the transmission of the first *partition* of the file. The file received incompletely prior to the re-transmission is discarded in the receiver.

In turn, a *TU command* is split into so-called *TU control blocks* transmitted within the *TU control block packet payload*. A *TU control block* must have a minimum length of 8 bytes unless it is the last (or the only) *TU control block* of a *TU command*.

If a CRC error is determined in the transmission of a *control payload* containing a *TU control packet*, the *TU control block* transmitted therein and, hence, the *TU command* containing this *TU control block* and, as appropriate, the complete file if transmitted in several *partitions* are discarded completely (see 7.4.2.6).

7.4.2.4.2 TU Command Format

The *TU command* structure is illustrated in Figure 7-9:

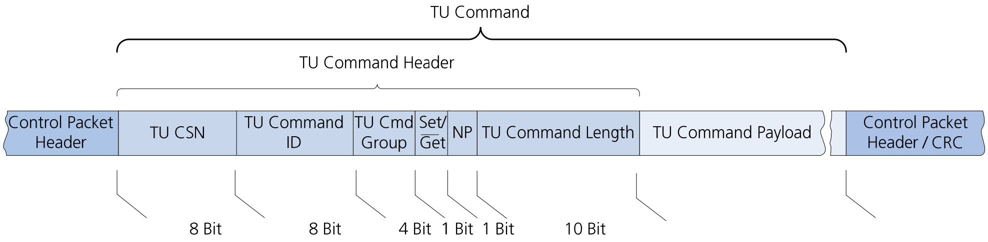


Figure 7-9: Command Structure of the TU Command

The components of the *TU command header* are described in Table 7-9.

Field	Description
TU CSN *Length*	The individual *TU commands* are numbered by means of the so-called *TU command sequence number*. *8 bits* If the counter reaches CSN 256 (0xFF), it re-starts with 0x00 at the next *TU command*. In this way, a number of 256 *TU commands* can be clearly referenced.
TU Command ID *Length*	The *TU command ID* defines which command is transmitted. *8 bits*
TU Command Group *Length*	By means of the *TU command group* individual *TU commands* can be grouped, e.g. TX, RX, BIT or peripheral components, such as power amplifier or ATU. *4 bits*
Set/$\overline{\text{Get}}$ *Length*	By means of the Set/$\overline{\text{Get}}$ bit it is signaled whether a value shall be set or read. *1 bit* 0 = Get 1 = Set
NP *Length*	*Next Partition* *1 bit* The NP bit is set when one or more *partitions* of a file (> 1 Kbyte) have already been transmitted in previous *TU commands*. 0 = partition 0 1 = partition 1 … n
TU Command Length *Length*	The length of the *TU command payload* is indicated in bytes by means of the *TU command length*. *10 bits* The *TU command length* can have values between 0 and 1020. Hence, the values 1021, 1022 and 1023 are excluded. Thus, the maximum length of a *TU command* including *TU command header* is 1024 bytes.

Table 7-9: TU Command Header Definition

Upon write access, the required data commensurate with the *TU command ID* and *TU command group* are transmitted within the *TU command payload*.

7.4.2.4.3 TU Application Response

No acknowledge to a *TU command* is transmitted on transport level. Each *TU command* correctly received retains a *TU application response*. It signals in its payload whether the *TU command* could have been interpreted correctly by the application. In case of errors, an indicator for the cause of the error is transmitted. If a Get command is received correctly (i.e. the Set/Get bit was not set), the parameter(s) read is/are additionally transmitted in the *TU application response payload*. If a *TU application response* cannot be transmitted within a *TU control packet payload* (e.g. in case of a larger file with a Get command), it must be split into *TU control blocks*.

The structure of this *TU application response* is almost identical to the structure of a *TU command* and retains the $\overline{\text{TU}}$ CSN of the *TU command* to which the response is transmitted. The *Set/$\overline{\text{Get}}$* bit, the *next partition* bit, the *TU command ID* and the *TU command group* of the previous *TU command* are also inherited into the *TU application response header*. The TU CSN in conjunction with the *TU command ID* and the *TU command group* is used to enable an association of the *TU application response* with the pertinent *TU command*.

At the beginning of the *TU application response payload* a 16-bit long *TU command acknowledge* is inserted signaling whether a correct *TU command* has been received or, in case not, what is causing the error.

The components of the *TU application response* are described in Table 7-10.

Field	Description
TU CSN	*TU command sequence number* of the *TU command* to which the *TU application response* is performed.
Length	*8 bits*
TU command ID	Here, the *TU command ID* of the *TU command* is transmitted to which the *TU application response* is performed.
Length	*8 bits*
TU command group	Here, the *TU command group* of the *TU command* is transmitted to which the *TU application response* is performed.
Length	*4 bits*
Set/$\overline{\text{Get}}$	The Set/$\overline{\text{Get}}$ bit is inherited immediately from the pertinent *TU command* and indicates whether this *TU command* was a write (Set) or a read access (Get).
	1 bit
Length	0 = Get 1 = Set
NP	*Next Partition*
Length	*1 bit* The NP bit is set when one or more *partitions* of a file (> 1 Kbyte) have already been transmitted in previous *TU commands*. 0 = partition 0

Field	Description
	1 = partition 1 … n
TU Application Response Length *Length*	The length of the payload of the *TU application response* is indicated in bytes by means of the *TU application response length*. *10 bits* In case of a Set command, the *TU application response length* is equal to "2", in case of a Get command it is greater than "2".
Payload *Length*	Initially, an acknowledge information with regard to the *TU command* is transmitted within the payload. In addition, the parameters requested are transmitted in case of a response to a Get command. The initial 16 bits of the payload signal whether the *TU command* could be interpreted error-free or, in case not, which error occurred (*TU command acknowledge*). 0x0000 = *TU command* free of errors 0x0001 = unknown *TU command* 0x0002 = *TU command* not defined in this operating condition 0x0003 = NP bit set incorrectly 0x0004 … 0x000F reserved for further error messages with regard to the OBISS interface 0x0010 … 0xFFFF available for application-specific error messages. Subsequently, a maximum of 1018 bytes of data follow in case of a Get command. In case of a Set command, the payload is composed of the *TU command acknowledge* only. *2 … 1020 bytes*

Table 7-10: TU Application Response Definition

7.4.2.4.4 Interface-specific TU Commands

[eventInfo] = getEventInfo( )

The getEventInfo command is used by the BBU to retrieve an event in the transceiver module in case the *event flag* has been set. If the event information cannot be transmitted in one 1024-byte *partition*, further *partitions* must be retrieved by repeating the command with the NP bit being then set.

eventInfo	provides information with regard to the event pending in the transceiver module in the form of a string with a maximum length of 1020 bytes.
TU CSN	is specified by the application
TU command ID	0x00

TU command group	0x0
Set/$\overline{\text{Get}}$	0 (= Get)
Next partition (NP)	0 if this is the first retrieval of the event 1 if one or more partitions have been transmitted to the event and more are pending
TU command length	0x000

clearEvent( )

The clearEvent command serves to signal to the transceiver module that the information with regard to an event has been successfully received and the event can be deleted in the transceiver module. As a result, the transceiver module deletes the *event flag* in the *message header* if there are no further events.

TU CSN	is specified by the application
TU command ID	0x01
TU command group	0x0
Set/$\overline{\text{Get}}$	1 (= Set)
Next partition (NP)	0
TU command length	0x000

setNewDataRate(dataMode)

The setNewDataRate command serves to signal to the transceiver module that the transmission over the interface shall be performed with another data rate supported by the transceiver module. Switching of the data rate and re-synchronization are conducted only if an error-free interpretation of the command is signaled within the *TU command acknowledge* of the *TU application response*. For further details of the data rate change commanded, see Chapter 7.3.4.

dataMode	determines the data rate and the IQ sample rate by means of which the communication between the BBU and the transceiver module shall take place (see Chapters 6.3.1 and 7.4.3). dataMode is a uint8 data type. For the association with the transmission modes and the pertinent data rates, see Table 7-11.
TU CSN	is specified by the application
TU command ID	0x02
TU command group	0x0
Set/$\overline{\text{Get}}$	1 (= Set)
Next partition (NP)	0
TU command length	0x001

dataMode	Designator	Description
1	Mode_A	Data rate i = 1: 768 MBit/s, IQ sample rate = 9.6 MSamples/s
2	Mode_B	Data rate i = 1: 1536 MBit/s, IQ sample rate = 19.2 MSamples/s
3	Mode_C	Data rate i = 1: 3072 MBit/s, IQ sample rate = 48 MSamples/s
4	Mode_D	Data rate i = 1: 3072 MBit/s, IQ sample rate = 72 MSamples/s

Table 7-11: dataMode Association with Data and IQ Sample Rate

abortFileTransfer(cmdId, cmdGrp)

This command serves to abort the transfer of a file transmitted in several *partitions*. In case of a Get command, it causes in the transceiver module the discarding of the provided *partitions* not yet transmitted. In case of a Set command, it causes the discarding of the *partitions* already received. If, subsequently, the command aborted with the abortFileTransfer command is re-activated, it re-starts with the first *partition*.

cmdId	specifies the *TU command ID* of the *TU command* to be aborted. cmdId is a uint8 data type.
cmdGrp	specifies the *TU command group* of the *TU command* to be aborted. cmdGrp is a uint8 data type with the 4 LSBs being used only. The 4 MSBs are transmitted in the transmitter with "0" and are ignored in the receiver.
TU CSN	is specified by the application
TU command ID	0x03
TU command group	0x0
Set/$\overline{\text{Get}}$	1 (= Set)
Next partition (NP)	0
TU command length	0x002

7.4.2.5 Padding (PD):

Padding data packets are used to fill up the *control payload* up to their size defined for the data rate used. The length of a *padding* data packet is always equal to 0. Consequently, only *PD control headers* are used for filling up the *control payload*.

7.4.2.6 Error Protection of the Control Payload

The *control payload* of each *message* is protected by a 16-bit CRC. The CRC is applied over the complete *control payload* including the *control packet header*.

$X^{16}+X^{12}+X^5+1$ is used as polynomial with the most significant bit (MSB) transmitted first. The CRC shift register is initialized by setting all elements to "1". Each *control payload* enters the shift register with the MSB first, i.e. with the first bit of the first *control packet header*.

If the receiver identifies a transmission error within the *control payload* by means of the CRC, all *TC commands* or *TU commands* transmitted completely or in parts within the *control payload* of this *message* are discarded.

7.4.3 Data Payload

The *data payload* of a *message* contains the IQ samples transported over the serial IQ baseband interface. The length of one *data payload* has been specified in Chapter 7.3.1 to 96, 192 and 480 bytes respectively. By default, the word length for one IQ sample is 48 bits, i.e. 6 bytes. With a fixed *message* length of 1.67 µs, the result is the following three sample rates:

- 9.6 MSamples / second (mode A)
- 19.2 MSamples / second (mode B) and
- 48 MSamples / second (mode C)

If waveforms shall be transmitted with maximum bandwidth, no higher dynamic range can be gained by sampling down in the receiver or sampling up in the transmitter since the data must be transmitted over the interface with the full sample rate. In this case, 16-bit word lengths for I and Q each and, hence, a complete word length of 32 bits per IQ sample are sufficient. Thus, a fourth transmission mode with reduced word length for an IQ sample of 32 bits was defined. Hence, the maximum sample rate with 480 bytes per *data payload* and 1.67 µs per *message* length is calculated to:

- 72 MSamples / second (mode D)

7.4.3.1 Data Payload Format

The formats for the IQ samples are specified as follows:

Standard IQ Sample in TX Direction
By default, 24 bits in 2-complement are transmitted for I and Q data each (i.e. 6 bytes for each IQ sample) in transmit direction with the MSB always being transmitted first. Figure 7-10 illustrates the way the 16 IQ samples for the lowest data rate (i = 1, commensurate with mode A) are embedded into the *data payload*. The same structure is used for the higher data rates (i = 2 and i = 4, commensurate with modes B and C) with 32 IQ samples and 80 IQ samples respectively contained in one *data payload* each.

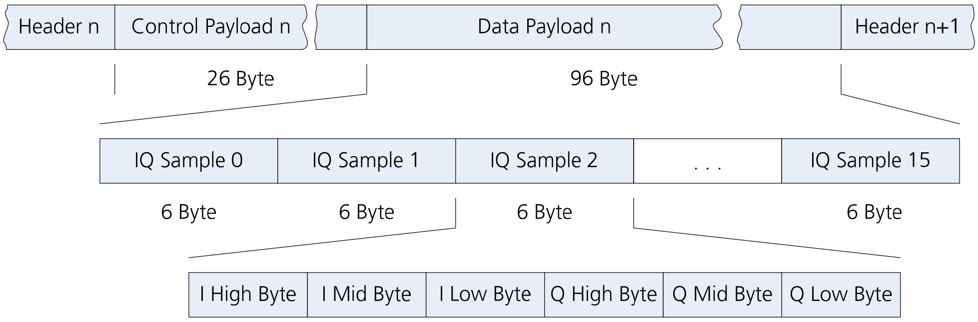


Figure 7-10: Mapping of IQ Data for TX within the Data Payload for the Case i = 1 (Mode A)

Standard IQ Sample in RX Direction

In receive direction, the IQ samples are transmitted exponentially with 22-bit mantissa (2-complement, MSB first) and 4-bit exponents with one common exponent for an IQ sample being transmitted only (i.e. 6 bytes for each IQ sample). Figure 7-11 illustrates the way the 16 IQ samples for the lowest data rate (i = 1, commensurate with mode A) are embedded into the *data payload*. The same structure is used for the higher data rates (i = 2 and i = 4, commensurate with modes B and C) with 32 IQ samples and 80 IQ samples respectively contained in one *data payload* each.

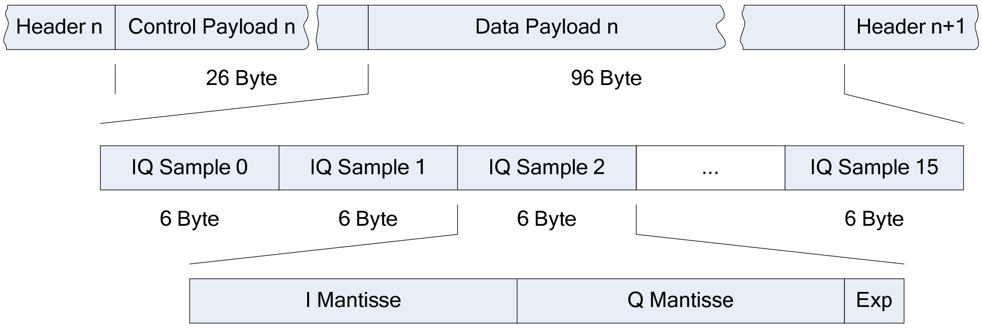


Figure 7-11: Mapping of IQ Data for RX within the Data Payload for the Case i = 1 (Mode A)

High Sample Rate of IQ Samples (in TX and RX Direction)

For very broad-banded waveforms, 16 bits in 2-complement for I and Q data each can be transmitted optionally (i.e. 4 bytes for each IQ sample) with the MSB being always transmitted first. This data format is available with the highest data rate (i = 4, then commensurate with mode D) only. Figure 7-12 illustrates the way the 120 IQ samples are embedded into the *data payload*.

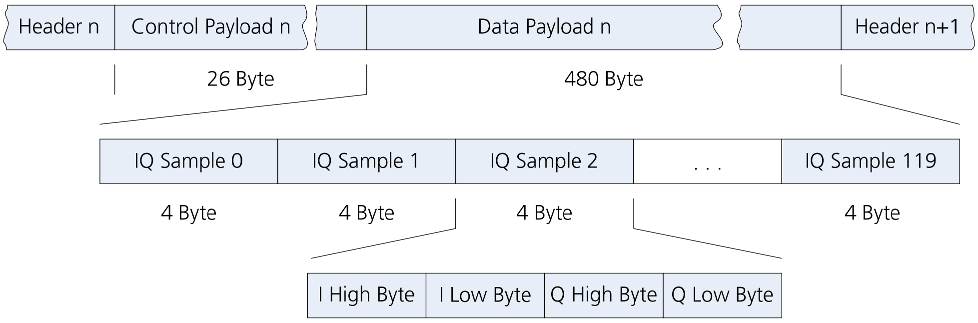


Figure 7-12: Mapping of IQ Data within the Data Payload for a Shorter Word Length Resulting in a Higher Sample Rate (Mode D)

7.4.3.2 Data Payload for Several Transmitters or Receivers on One Transceiver Module

The samples within the *data payload* can be distributed to several transmitters or receivers in one transceiver module. For this purpose, 1, 2, 4 or 8 transmitters or receivers are intended per module. This, for example, allows applications with multiple antenna systems (e.g. beam forming, MIMO – also see Chapter 6).

The number of transmitters does not have to correspond to the number of receivers on the transceiver module. This means that the format of the *data* *payload* can be defined in different ways for TX and RX (see Chapter 7.4.1, *data type DT*, also see Figure 6-2).

During the initialization, the transceiver module must communicate the number of transmitters and receivers on the module to the BBU (default setting is one transmitter and one receiver on the transceiver module).

The order of the samples of the *Data Payload* with n transmitters and n receivers respectively, is specified in such a way that the first sample for transmitter 0 or from receiver 0 is transmitted, followed by the first sample for transmitter 1 or from receiver 1 until the first sample for transmitter n or from receiver n is transmitted, then the second sample for transmitter 0 or from receiver 0 is transmitted etc. (see Figure 7-13).

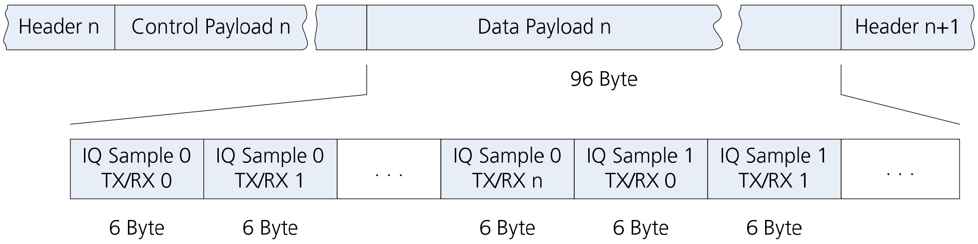


Figure 7-13: Mapping of IQ Data for n Transmitters / Receivers within the Data Payload for the Case i = 1 (Mode A)

7.4.3.3 Reduced Sample Rate over the Data Payload

The data rate to the transmitter or from the receiver of the transceiver module can be further reduced by means of the definition of the *data type* in the *message header* of the *message*.

For this purpose, more transmitters or receivers are defined in the *data type* in TX or RX direction than are available on the transceiver module.

Example:

In mode A (data transmission rate over the interface is 768 Mbit/s), one *data payload* contains 96 bytes or 16 samples, the maximum sample rate is set to 9.6 MSamples/s (if bit 4 has been set in DT).

If bit 6 is set instead of bit 4 in DT although only one transmitter or receiver is available on the transceiver module, 4 of the 16 samples (samples 0, 4, 8 and 12) are transferred to the transmitter in the transceiver module or to the receiver in the BBU only, the remaining bits are discarded. This way, the sample rate is reduced by factor 4 to 2.4 MSamples/s. Figure 7-14 clarifies this example.

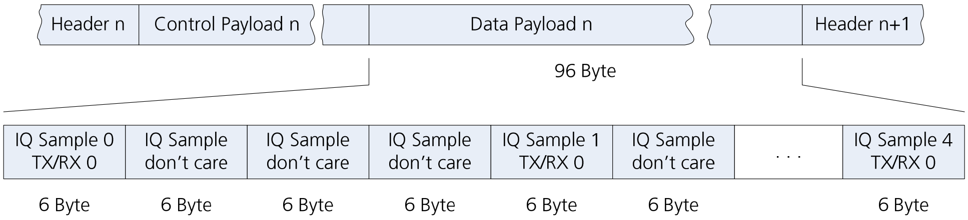


Figure 7-14: Mapping of IQ Data within the Data Payload when Reducing the Data Rate by Factor 4 for the Case i = 1 (Mode A)

The possible sample rates in the *data payload* are:

Mode A:

DT=0x11 ⇒ 9.6 MSamples/s, DT=0x21 ⇒ 4.8 MSamples/s, DT=0x41 ⇒ 2.4 MSamples/s, DT=0x81 ⇒ 1.2 MSamples/s

Mode B:

DT=0x11 ⇒ 19.2 MSamples/s, DT=0x21 ⇒ 9.6 MSamples/s, DT=0x41 ⇒ 4.8 MSamples/s, DT=0x81 ⇒ 2.4 MSamples/s

Mode C:

DT=0x11 ⇒ 48 MSamples/s, DT=0x21 ⇒ 24 MSamples/s, DT=0x41 ⇒ 12 MSamples/s, DT=0x81 ⇒ 6 MSamples/s

Mode D:

DT=0x13 ⇒ 72 MSamples/s, DT=0x23 ⇒ 36 MSamples/s, DT=0x43 ⇒ 18 MSamples/s, DT=0x83 ⇒ 9 MSamples/s

8 Synchronization

8.1 Time Synchronization

Time synchronization is the ability to synchronize temporal reference systems, perform actions at defined points in time and/or to assign a point in time to past events. For this purpose, a point in time is defined as specific moment in a temporal reference system.

When using the IQ baseband interface, time synchronization can be used for

- the transmission of TX samples at specified points in time
- the temporally assignable receipt of RX samples
- the activation and de-activation of receivers or transmitters and/or the switching between transmitter and receiver at specified points in time
- the switching of frequencies for transmitters and receivers at specified points in time
- for AGC control at specific points in time,

among others.

The time synchronizations mentioned below will be defined in the following:

- time synchronization of the transceiver module time with the BBU time
- signal time synchronization of TX samples
- signal time synchronization of RX samples
- action time synchronization

8.1.1 Time Synchronization of the Transceiver Module Time with the BBU Time

The BBU time and the transceiver module time are defined as temporal reference systems.

For this purpose, the BBU time designates the temporal reference system of the BBU with regard to the IQ baseband interface. The transceiver module time designates the temporal reference system of the transceiver module with regard to the IQ baseband interface.

The synchronization of the transceiver module time with the BBU time is performed by the first bit of *TX messages* transmitted from the BBU to the transceiver module.

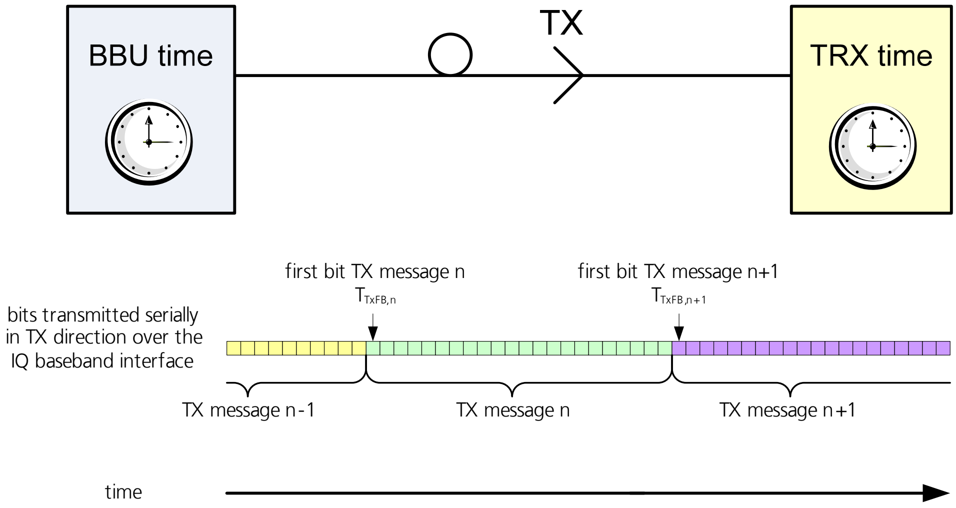


Figure 8-1: Synchronization of Transceiver Module Time with Regard to BBU Time

The achievable temporal accuracy of the synchronization of the transceiver module time with the BBU time depends on the BBU's temporal accuracy (with regard to the BBU time) to provide the first bit of a *message* to the IQ baseband interface input of the transceiver module and the transceiver module's temporal accuracy to detect the first bit of a *message*. In the following, this accuracy is designated as time synchronization accuracy A_{TBS}.

By this procedure, the transceiver module can, in theory, be synchronized with the BBU time with a definiteness of up to a maximum of one second. For a definite time reference of more than one second, additional information must be transmitted from the BBU to the transceiver module.

8.1.2 Signal Time Synchronization of TX Samples

The signal time synchronization allows the transmission of a signal in the transmit path at a defined point in time.

The transmit time of a sample is the point in time at which 50% of the energy assigned to the sample is transmitted at the antenna connector of the transceiver module.

The signal time synchronization in the transmit path is based on a defined TX latency of the transceiver module designated as D_{TxL}. As shown in Figure 8-2, it is defined as the period between the arrival of the first bit of a *message* T_{TxFB} at the interface of the transceiver module and the transmit time T_{TxFS} of the first sample of this *message* at the antenna connector.

$$D_{TxL} = T_{TxFS,n} - T_{TxFB,n}$$

Equation 8-1

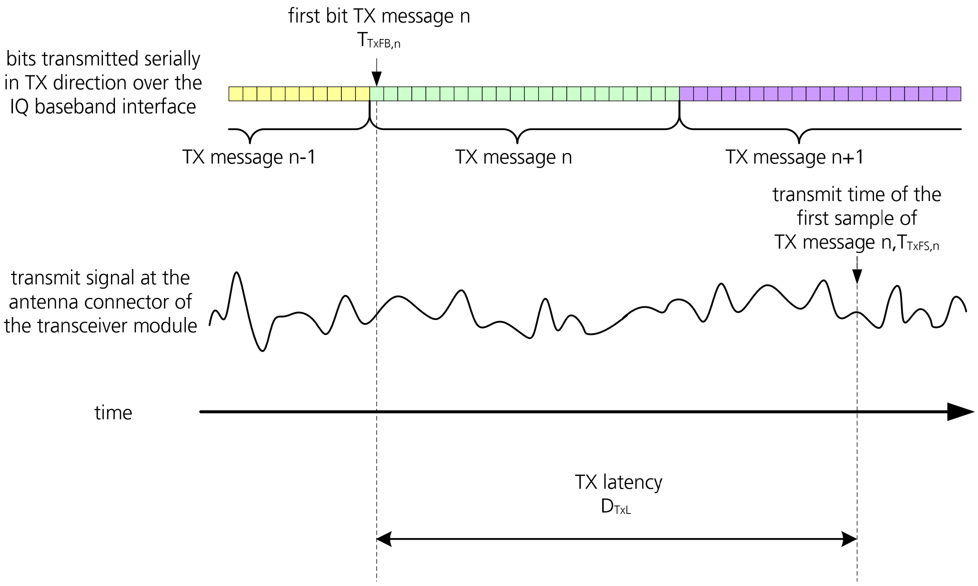


Figure 8-2: Signal Time Synchronization in the Transmit Path

TX latency may depend on the configuration (e.g. the group delay time of the digital and analog filters in the transceiver module). It must be possible to predetermine the latency for a defined configuration and it must be made available to the application. For a temporally defined transmission of TX samples, the BBU must take this TX latency into account (plus the duration over the interface from the BBU to the transceiver module, if required).

If the first TX sample of *TX message* n shall be transmitted at time $T_{TxFS,n}$ with regard to the BBU time, the BBU must ensure that the first bit of *message* n arrives at time

$$T_{TxFB,n} = T_{TxFS,n} - D_{TxL} \qquad \text{Equation 8-2}$$

with regard to the BBU time at the interface of the transceiver module.

Then, the first TX sample of *message* n is transmitted with regard to the BBU time at time

$$T_{TxFS,n} = T_{TxFB,n} + D_{TxL} \qquad \text{Equation 8-3}$$

With regard to the BBU time, the accuracy of transmit time $T_{TxFS,n}$ depends on time synchronization accuracy A_{TBS}.

8.1.3 Signal Time Synchronization of RX Samples

The signal time synchronization allows the association of a received signal in the receive path with a specific point in time.

The time of receipt of a sample is that point in time at which 50% of the energy associated with the sample has been received at the antenna connector of the transceiver module.

The samples are temporally referred to (virtual) transmit times of TX samples at the antenna connector of the same transceiver module in order to associate as precisely as possible the points in time of signals received with the points in time of signals transmitted. The term "virtual" refers to the fact that usually no actual transmit operations are performed; however, when receiving, *messages* are continuously transmitted over the IQ baseband interface.

The signal time synchronization in the transmit path is based on a defined TX/RX latency of the transceiver module designated as D_{TxRx}. As illustrated in Figure 8-3, it is calculated as period between time of receipt $T_{RxFS,n}$ of the first sample of *RX message* n and virtual transmit time $T_{TxFS,n}$ of the first sample of *TX message* n.

$$D_{TxRx} = T_{TxFS,n} - T_{RxFS,n} \qquad \text{Equation 8-4}$$

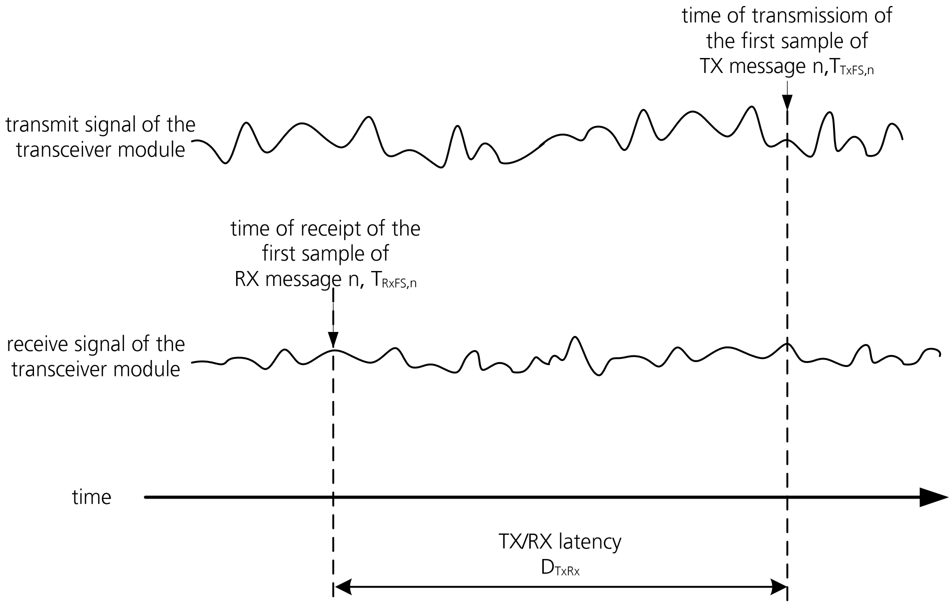


Figure 8-3: Signal Time Synchronization in the Receive Path in Relation to the Time of Transmission of a TX Sample

Due to the offset of *RX messages* to *TX messages*, D_{TxRx} can, in theory, also have negative values up to *message* length.

TX/RX latency depends on the configuration (e.g. the group delay time of the digital and analog filters in the transceiver module). It must be pre-determined for a configuration defined and must be made available to the application.

By means of

$$T_{RxFS,n} = T_{TxFS,n} - D_{TxRx} \qquad \text{Equation 8-5}$$

a determination of the time of receipt of an RX sample with regard to the (virtual) transmit time of a TX sample is possible—not dependent on the time synchronization accuracy A_{TBS}.

With regard to the BBU time, time of receipt $T_{RxFS,n}$ of the first sample of *RX message* n is calculated with Equation 8-3 to

$$T_{RxFS,n} = T_{TxFB,n} + D_{TxL} - D_{TxRx} \qquad \text{Equation 8-6}$$

For this purpose, $T_{TxFB,n}$ is the time of the first bit of *message* n coming in at the device interface of the transceiver module. With regard to the BBU time, the accuracy of transmit time $T_{TxFS,n}$ depends on time synchronization accuracy A_{TBS}.

Transmit/receive delay N_{TxRx} is defined as period (in number of samples) between a signal in the TX samples to the pertinent own receive signal in the RX samples in case of an overcoupling at the antenna connector.

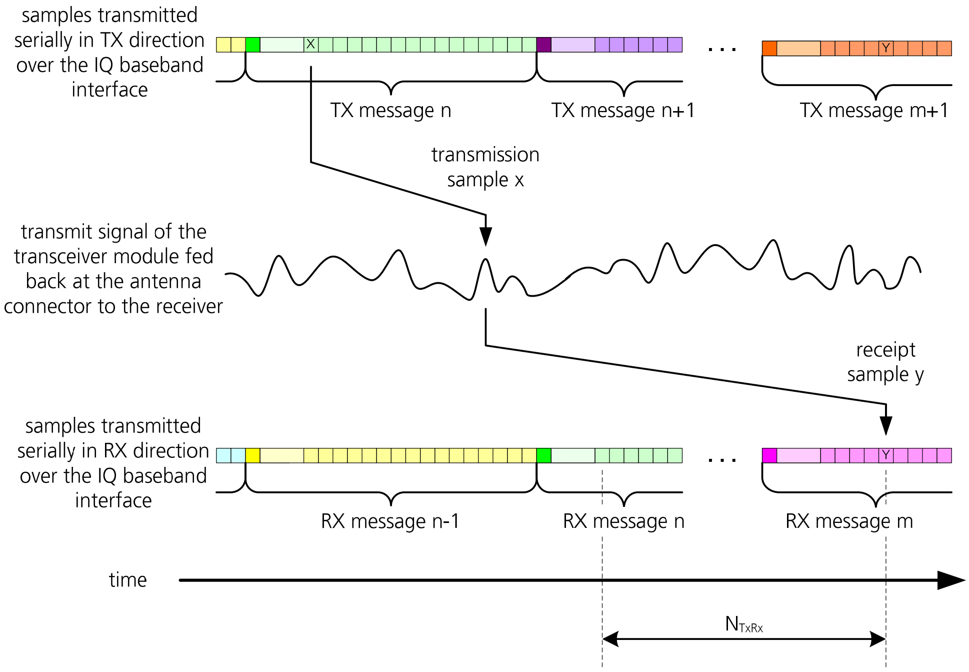


Figure 8-4: Transmit/Receive Delay

With knowledge of the sampling rate f_{samp} of the interface N_{TxRx} can be derived:

$$N_{TxRx} = D_{TxRx} * f_{samp} \qquad \text{Equation 8-7}$$

N_{TxRx} is independent from time synchronization accuracy A_{TBS}.

8.1.4 Action Time Synchronization

By means of the action time synchronization, actions can be performed in temporal reference to TX samples, to RX samples or to points in time (e.g. frequency hopping or transmitter/receiver switching).

For this purpose, the IQ baseband interface offers the option to signal a TX sample as reference point in each *TC command*. As it will be explained in the following chapters, TX sample reference points can be converted to RX sample reference points and time reference points.

By signaling a reference point alone, no time sequence of the action has been defined yet, e.g. when the action starts and when it ends. The time sequence of an action with regard to the reference point must be defined when defining the individual *TC command*.

The Annex (Chapter 14.3) contains illustrations of the action time synchronization.

8.1.4.1 TX Sample Reference Point

TX sample reference point $S_{RefTx,A}$ of an action is defined as that TX sample that follows N_{OR} samples after the first TX sample of the *message* in which the action is signaled in a *TC command*.

Reference to a TX sample reference point is independent from time synchronization accuracy A_{TBS}.

Figure 8-5 illustrates signaling of TX sample reference point $S_{RefTx,A}$ in a *TC command*. Action A is signaled in a *TC command* of the *control payload* of *TX message* n.

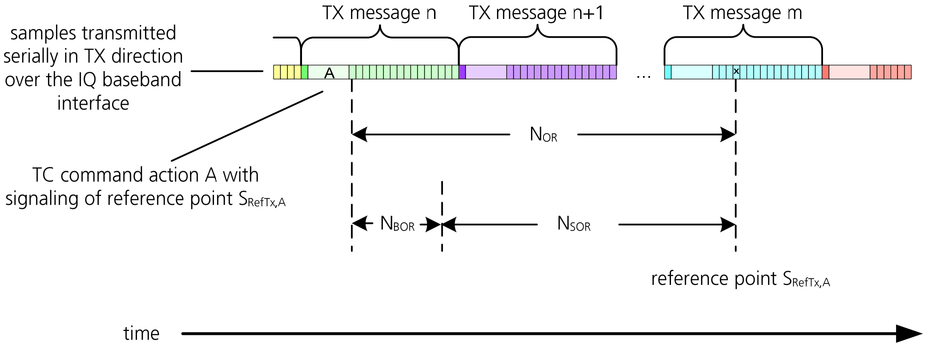


Figure 8-5: TX Sample Reference Point

N_{OR} is the number of TX samples transmitted between the *control payload*, within which the command is transmitted, and the TX sample that is the TX sample reference point of the action.

The N_{OR} value is composed of an N_{SOR} value made known by the transceiver module and an N_{BOR} value transmitted in the respective *TC command* by the BBU.

The N_{SOR} value is made known to the BBU by the transceiver module.

The N_{BOR} value is transmitted in the respective *TC command* from the BBU.

Hence, the following applies

$$N_{OR} = N_{BOR} + N_{SOR} \qquad \text{Equation 8-8}$$

$$N_{BOR} \in [0\ldots4094] \qquad \text{Equation 8-9}$$

In the transceiver module, different actions may require different periods between signaling the action over the device interface and performing the action. When determining the N_{SOR} value, all times in the transceiver module must be taken into account so that an action with $N_{BOR} = 0$ can be performed in good time.

If an action shall be performed at a specific time, the BBU must take the N_{SOR} value into account when determining the *message* in which the *TC command* with the action is transmitted as well as when determining the N_{BOR} value.

8.1.4.2 RX Sample Reference Point

TX sample reference points can be converted to RX sample reference points as per the following definition:

If TX sample reference point $S_{RefTx,A}$ corresponds to sample x of *TX message* n, RX sample reference point $S_{RefRx,A}$ corresponds to RX sample y which follows N_{TxRx} samples after RX sample x of *RX message* n.

This relationship is illustrated in Figure 8-6.

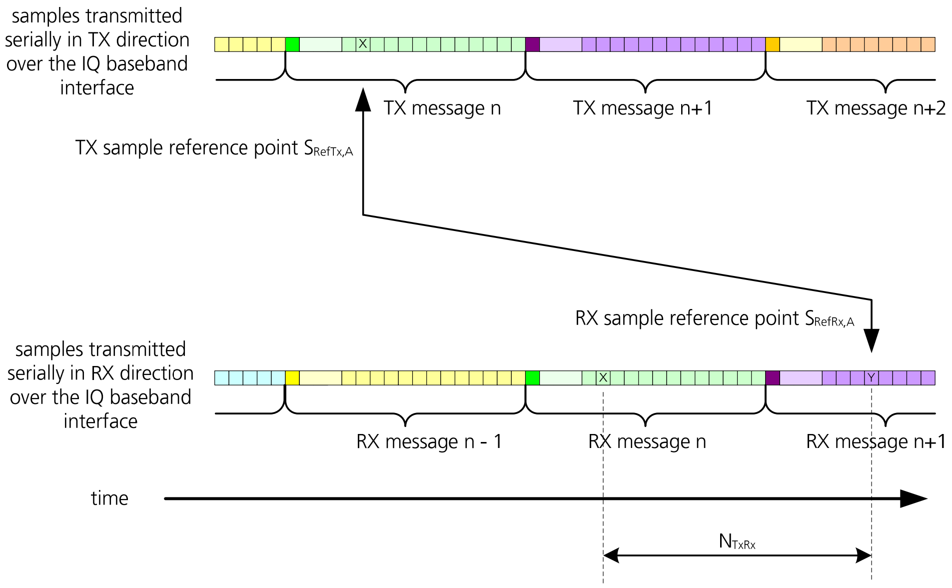


Figure 8-6: Association of an Action with an RX Sample

Reference to an RX sample reference point is independent from time synchronization accuracy A_{TBS}.

8.1.4.3 Time Reference Point

TX sample reference points can be converted to time reference points as per the following definition:

If an action is signaled in a *TC command* of the *control payload* of *TX message* n, the signaled BBU point in time can be calculated to

$$T_{RefAnt,A} = T_{TxFB,n} + D_{TxL} + N_{OR} / f_{samp} \qquad \text{Equation 8-10}$$

This is the point in time when (virtual) TX sample reference point $S_{RefTx,A}$ is transmitted at the antenna connector.

f_{samp} has been defined as sample rate of the interface.

Reference to the BBU time is dependent on time synchronization accuracy A_{TBS}.

8.2 Frequency Synchronization

The IQ baseband interface shall enable the transceiver module to keep high frequency accuracy without complying with its own highly accurate frequency standard.

The transceiver module shall, for example, synchronize its reference clock for the frequency generation required (e.g. for sampling, local oscillators as well as the bit clock for the interface in receive direction) via a PLL to the bit clock of the IQ baseband interface.

9 List of References

[5] A. X. Widmer, P. A. Franszek, A DC-Balanced, Partitioned-Block, 8B/10B Transmission Code, IBM Journal of Research and Development, Vol. 27, No. 5, pp. 440-451, Sep 1983

[6] IEEE Standard for information technology – Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications, IEEE Std. 802.3-2005

[7] IEEE: Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications, Amendment: Media Access Control (MAC) Parameters, Physical Layers, and Management Parameters for 10 GB/s Operation, IEEE Std. 802.3ae-2002, Mar 2002

10 List of Figures

11 List of Tables

12 List of Abbreviations

AD	Analog/Digital
AGC	Automatic Gain Control
ATU	Antenna Tuning Unit
BBU	Baseband Unit
CDMA	Code Division Multiple Access
CMD	Command
CP	Control Payload
CPRI	Common Public Radio Interface
CRC	Cyclic Redundancy Check
CSN	Command Sequence Number
DA	Digital/Analog
DAC	Digital/Analog Converter
DC	Don’t Care (= bit with no information content in the *control payload*)
DDC	Digital Downconversion
DP	Data Payload
DSP	Digital Signal Processor
DUC	Digital Upconversion
FEC	Forward Error Correction
FIFO	First In First Out
FPGA	Field Programmable Gate Array
GSM	Global System for Mobile Communications
HF	High Frequency
I	In-phase
IBIT	Initiated Built-in Test
IF	Intermediate Frequency
INT	Interface
MIMO	Multiple In Multiple Out
MSB	Most Significant Bit
OBISS	Open Baseband Interface Specification for SDR
OBIT	Operational Built-in Test
OBSAI	Open Base Station Architecture Initiative
PHY	Physical Layer
PLL	Phase Locked Loop
Q	Quadrature-phase
Resp	Response
RFU	Reserved for Future Use

RX	Receive
SBIT	Start-up Built-in Test
SDR	Software Defined Radio
TRX	Transceiver
TX	Transmit
UMTS	Universal Mobile Telecommunications System

13 List of Symbols

A_{TBS}	Accuracy Time BBU System – time synchronization accuracy (Chapter 8.1.1)
BLOCK_SIZE	number of bytes within one block (Chapter 7.3.2 - frame synchronization)
CEF	CRC Error Flag (Table 7-5)
CSN	Command Sequence Number (Table 7-7 through Table 7-10)
DT	Data Type (Table 7-5)
D_{delay}	fixed, application-dependent delay of a transmit signal after receipt of a signal for transit time measurement (Chapter 14.3.1.4)
$D_{IQ(DSP)}$	signal transit time of a TX sample in the digital signal processing unit including DAC (Figure 14-16)
$D_{IQ(IF)}$	signal transit time of a TX sample in the ZF processing unit (Figure 14-16)
$D_{IQ(INT)}$	signal transit time of a TX sample in the interface (Figure 14-16)
$D_{IQ(PA)}$	signal transit time of a TX sample in the power amplifier (Figure 14-16)
$D_{IQ(UPCONV)}$	signal transit time of a TX sample in the upconverter (Figure 14-16)
D_{LZ}	signal transit time from transmitter to receiver (*Equation 14-4*)
$D_{pretune}$	time required prior to the actual frequency change to, for example, program the synthesizer (Chapter 14.3.2.1.2)
D_{RxL}	Delay RX Latency – period between the time of receipt of the first sample of the message at the antenna connector and the time of the first bit of a message coming in at the interface (Chapter 14.3)
$D_{TC(INT)}$	signal transit time for a TC command in the interface (Figure 14-16)
D_{TRM}	Delay TX-RX Message – delay between TX and RX message (Chapter 7.3.1)
D_{tune}	time required for switching the frequency (response time of the synthesizers, adjustment time of the filters - Chapter 14.3.2.1.2)
D_{TxL}	Delay TX Latency – period between the first bit of a message coming in at the interface and the transmission of the first sample of the message at the antenna connector (Equation 8-1)
D_{TxRx}	Delay TX RX – period between receipt of the first sample of RX message n and transmission of the first sample of TX message n (*Equation 8-4*)
EF	Event Flag (Table 7-5)

f_{old}	frequency prior to switching frequencies in case of frequency hopping (Chapter 14.3.2.1.1)
FCTU	Flow Control Time Uncritical Control Data (Table 7-5)
f_{new}	frequency after switching frequencies in case of frequency hopping (Chapter 14.3.2.1.1)
f_{samp}	sampling rate of the interface (Chapter 8.1.3)
IDLE	state in the state diagram of the transmitter (Figure 7-3)
LOF	Loss of Frame – is set when the receiver changes from SYNC state into WAIT_FOR_K28.1_SYNC state (Table 7-2)
LOF_EN	LOF Enable - indicates whether the "Loss of Frame" signal has an impact on the state of the transmitter (Table 7-2)
LOS	Loss of Signal – is set when the receiver is in UNSYNC state (Table 7-2)
LOS_EN	LOS Enable - indicates whether the "Loss of Signal" signal has an impact on the state of the transmitter (Table 7-2)
MaxRxTime	maximum receive time in case of a specific data rate (Table 7-3 - automatic negotiation of the data rate)
MaxTxTime	maximum transmit time in case of a specific data rate (Table 7-4 - automatic negotiation of the data rate)
MC	Message Counter (Table 7-5)
MC_{RefTx}	number of the message (in the message counter) with the TX sample reference point of the action (Chapter 14.3.2.2.2)
MC_{SOR}	number of the message (in the message counter) within which the TC command must be transmitted at the latest so that the transceiver module can still perform the command (Chapter 14.3.2.2.2)
NewDataRate	indicates whether a new data rate was set (Table 7-4 - automatic negotiation of the data rate)
NBOR	action delay in number of samples (commensurate with N_{BOR} - Table 7-7 and Table 7-8)
N_{BOR}	BBU Offset Reference – BBU-dependent portion of the number of samples from the TC command to the sample an action has been associated with (Chapter 8.1.4.1)
$N_{Message}$	number of samples per message (Chapter 14.3.2.1.3)
N_{OR}	Offset Reference – number of samples from the TC command to the sample an action has been associated with (Chapter 8.1.4.1)
NP	Next Partition bit (Table 7-9 and Table 7-10)
N_{SOR}	Transceiver Module Offset Reference – transceiver module-dependent portion of the number of samples from the TC command to the sample an action has been associated with (Chapter 8.1.4.1)
N_{TH}	hold time of an action – the signal path must not change within this period (Chapter 14.3.2.1.1)

N_{TS}	setup time of an action – the action must be completed within this period (Chapter 14.3.2.1.1)
N_{TxRx}	number of samples within period D_{TxRx} (*Equation 8-7*)
OFF	state in the state diagram of the transmitter (Figure 7-3)
PD	Padding (Chapter 7.4.2.5)
PV	Protocol Version = version number of this specification (Table 7-5)
SB	Status Bits (Table 7-5)
SCB	Serial Communication Bit (Table 7-5)
SOM	Start of Message (Table 7-5)
$S_{RefRx,A}$	Sample Reference RX, Action – RX sample reference point of an action (Chapter 8.1.4.2)
$S_{RefTx,A}$	Sample Reference TX, Action – TX sample reference point of an action (Chapter 8.1.4.1)
SYNC	state in the state diagram of the receiver
SYNC_M	number of bytes to the next sync code in SYNC state (Table 7-2)
SYNC_T	threshold value for consecutive valid blocks in order to change into WAIT_FOR_K28.1_SYNC state (Table 7-2)
UNSYNC	state in the state diagram of the receiver (Figure 7-4)
UNSYNC_T	threshold value for consecutive invalid blocks in order to change into UNSYNC state (Table 7-2)
TC	time critical (Chapter 7.4.2.3)
$T_{RefAnt,A}$	Time Reference at Antenna, Action – point in time at which the TX sample reference point at the antenna connector is transmitted (*Equation 8-10*)
$T_{RxFB,n}$	Time RX First Bit, Message n – point in time of the first bit of RX message n at the interface of the transceiver module (Chapter 14.3.1.1)
$T_{RxFB,BBE,n}$	Time RX First Bit at BBE, Message n – point in time of the first bit of RX message n at the interface of the BBU (Chapter 14.3.1.2.2)
$T_{RxFS,n}$	Time RX First Sample, Message n – point in time of the first sample of RX message n at the antenna connector (*Equation 8-5*)
$T_{TxFB,n}$	Time TX First Bit, Message n – point in time of the first bit of TX message n at the interface of the transceiver module (*Equation 8-2*)
$T_{TxFB,BBE,n}$	Time TX First Bit at BBE, Message n – point in time of the first bit of TX message n at the interface of the BBU (Chapter 14.3.1.2.2)
$T_{TxFS,n}$	Time TX First Sample, Message n – point in time of the first sample of TX message n at the antenna connector (*Equation 8-3*)

TU	time uncritical (Chapter 7.4.2.4)
TX_BUSY	state in the state diagram of the transmitter (Figure 7-3)
TX_EN	TX Enable - parameter is set in case transmissions shall be conducted over the interface (Table 7-2)

14 Annex

14.1 TC Command

The following figure illustrates the way a *TC command* is assembled into a *TC control packet* and inserted into the *control payload* of a *message*.

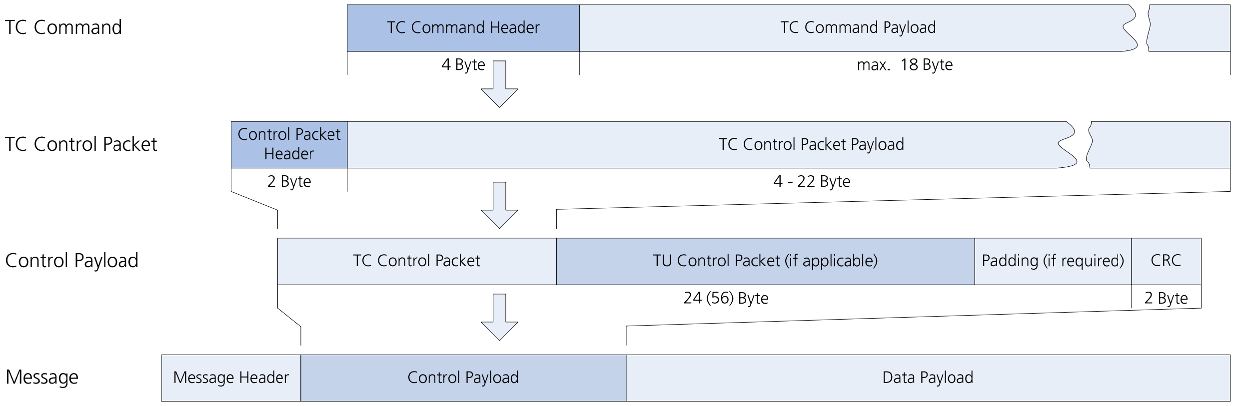


Figure 14-1: Insertion of a TC Command into the Control Payload

14.2 TU Command

The following figure illustrates the way a larger file is split into partitions and how these, together with the *TU command header*, are packed into single *TU control packets* and inserted into the *control payload* of *messages*.

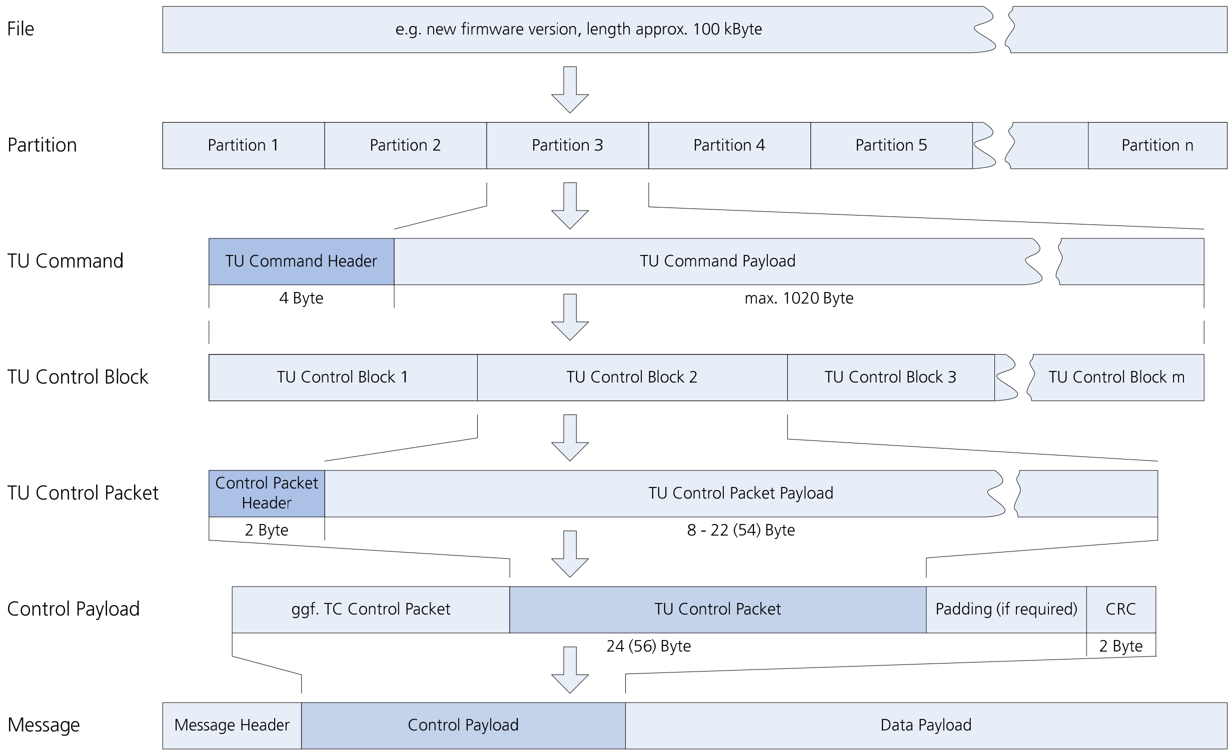


Figure 14-2: Insertion of a TU Command into the Control Payload

14.2.1 Sequence Diagram for a Set TU Command

Example: By means of the setTxPower command, the BBU sets a new output power in the transceiver module and passes an 8-bit TxPower value. The data to be transmitted within the payload of the *TU command* have a length of M = 1 byte.

$M = 1 = m * 1020 + k$

Hence, m = 0 and k =1 applies.

This specification does not provide any information about the application layer of the BBU. For example, it leaves open whether an acknowledge coming from the transceiver module is transferred to the application layer of the BBU or not.

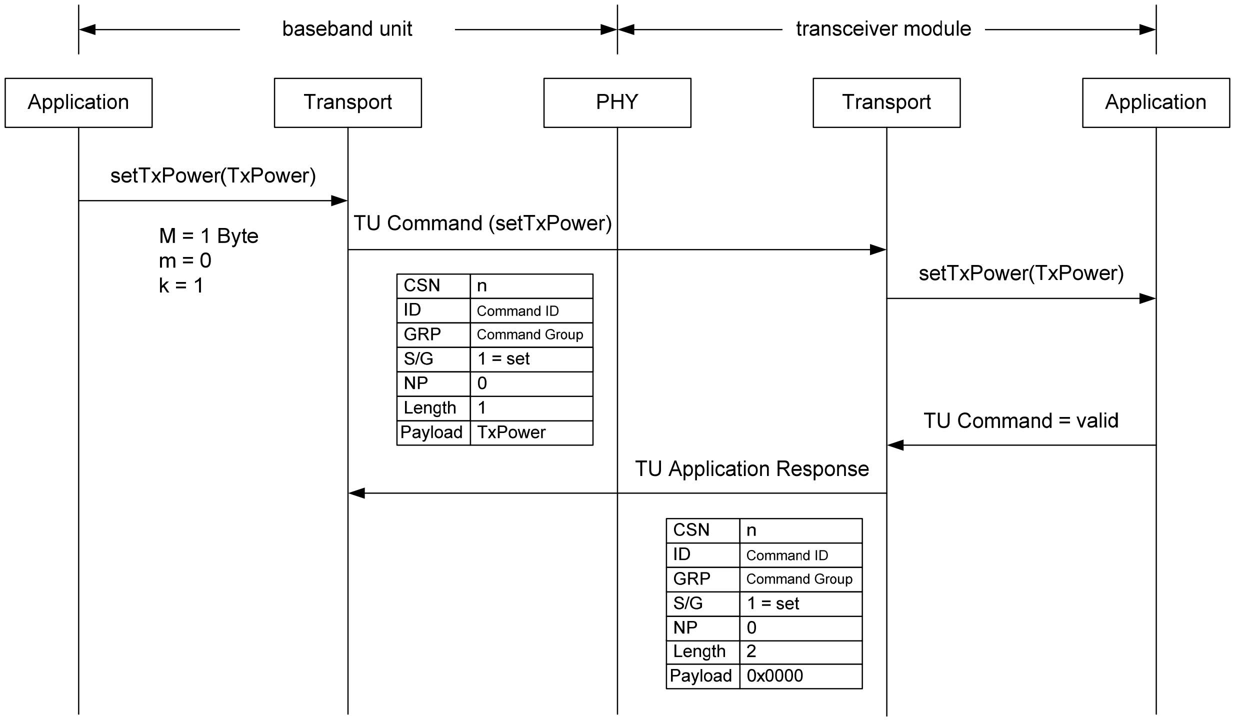


Figure 14-3: Sequence Diagram for a Set TU Command

14.2.2 Sequence Diagram for a Get TU Command

Example: The BBU intends to read out a configuration file from the transceiver module. The data to be transmitted within the payload of the *TU command* have a length of M = 0 bytes.

$M = 0 = m * 1020 + k$

Hence, m = 0 and k = 0 applies.

The configuration file has a length of N = 2650 bytes.

$N = 2650 \text{ bytes} = j * 1018 + i.$

Hence, j = 2 and i = 614 applies.

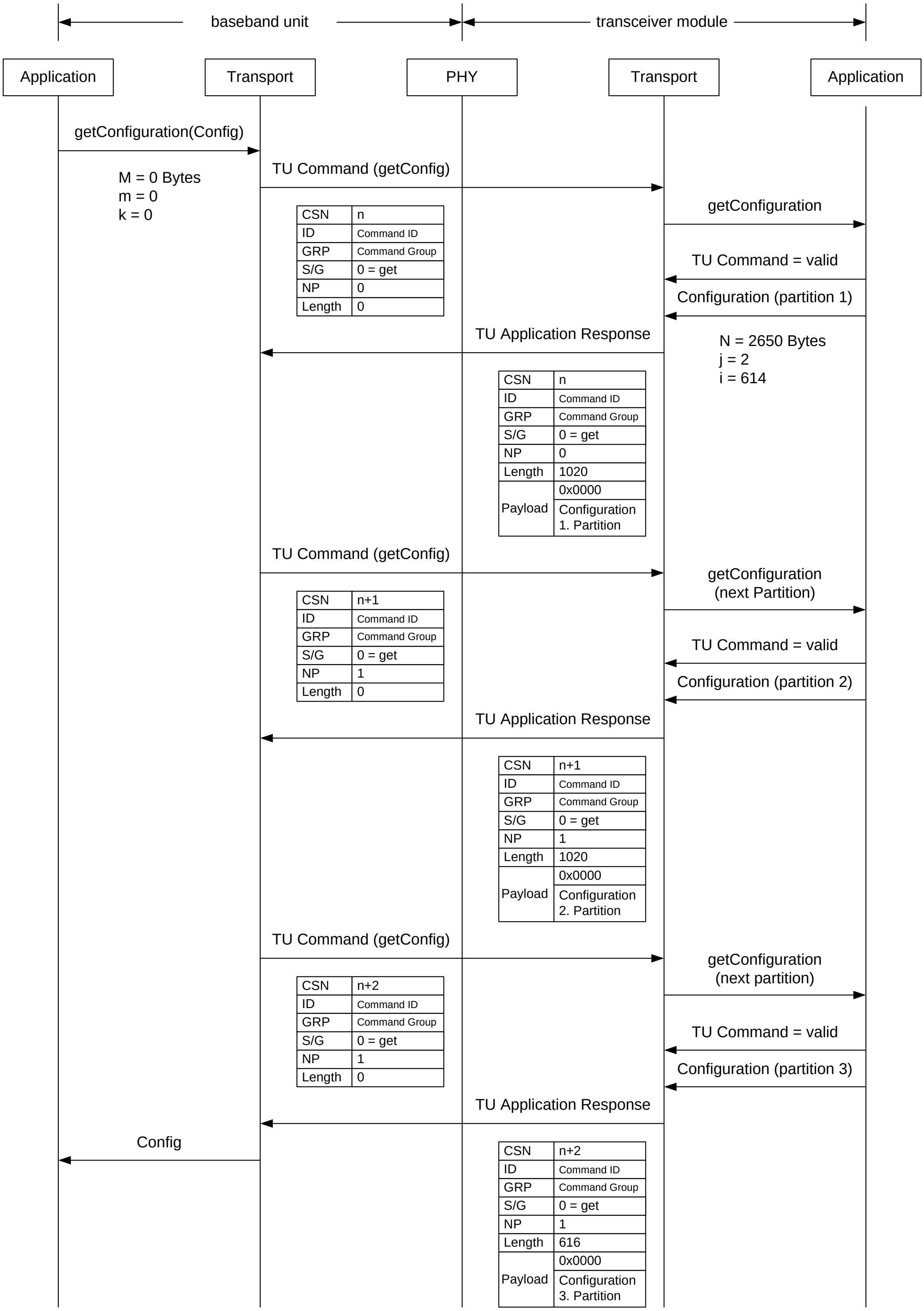


Figure 14-4: Sequence Diagram for a Get TU Command

14.2.3 Sequence Diagram for a Get TU Command with the File Incompletely Transmitted Being Re-requested

Example: The BBU intends to read out a configuration file from the transceiver module. The data to be transmitted within the payload of the *TU command* have a length of M = 0 bytes.

$M = 0 = m * 1020 + k$

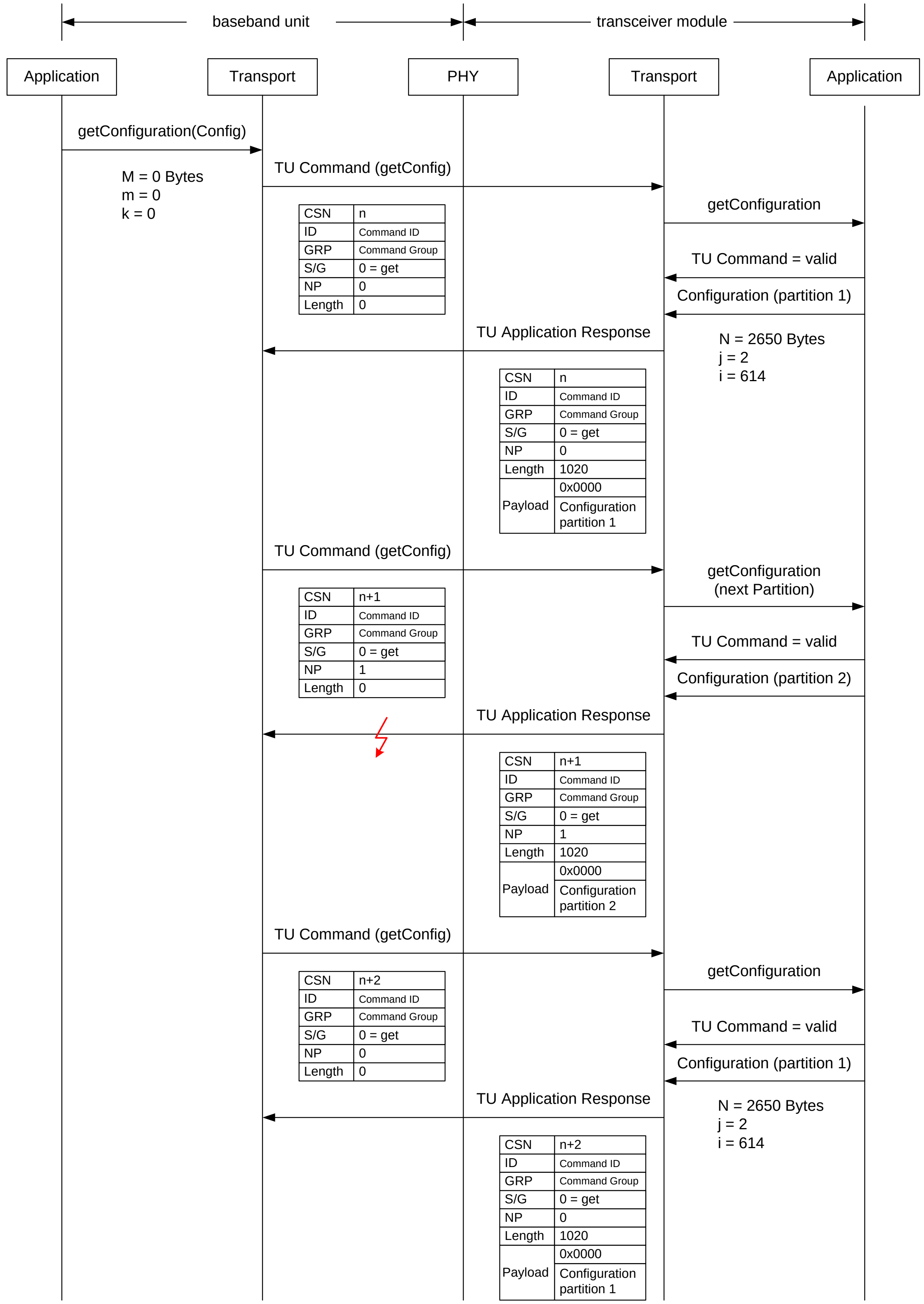
Hence, m = 0 and k = 0 applies.

The configuration file has a length of N = 2650 bytes.

$N = 2650 \text{ bytes} = j * 1018 + i.$

Hence, j = 2 and i = 614 applies.

Due to a transmission error, the second partition has not been correctly received in the BBU. Thus, the configuration file is re-requested.



CSN	n
ID	Command ID
GRP	Command Group
S/G	0 = get
NP	0
Length	0

CSN	n
ID	Command ID
GRP	Command Group
S/G	0 = get
NP	0
Length	1020
Payload	0x0000
	Configuration partition 1

CSN	n+1
ID	Command ID
GRP	Command Group
S/G	0 = get
NP	1
Length	0

CSN	n+1
ID	Command ID
GRP	Command Group
S/G	0 = get
NP	1
Length	1020
Payload	0x0000
	Configuration partition 2

CSN	n+2
ID	Command ID
GRP	Command Group
S/G	0 = get
NP	0
Length	0

CSN	n+2
ID	Command ID
GRP	Command Group
S/G	0 = get
NP	0
Length	1020
Payload	0x0000
	Configuration partition 1

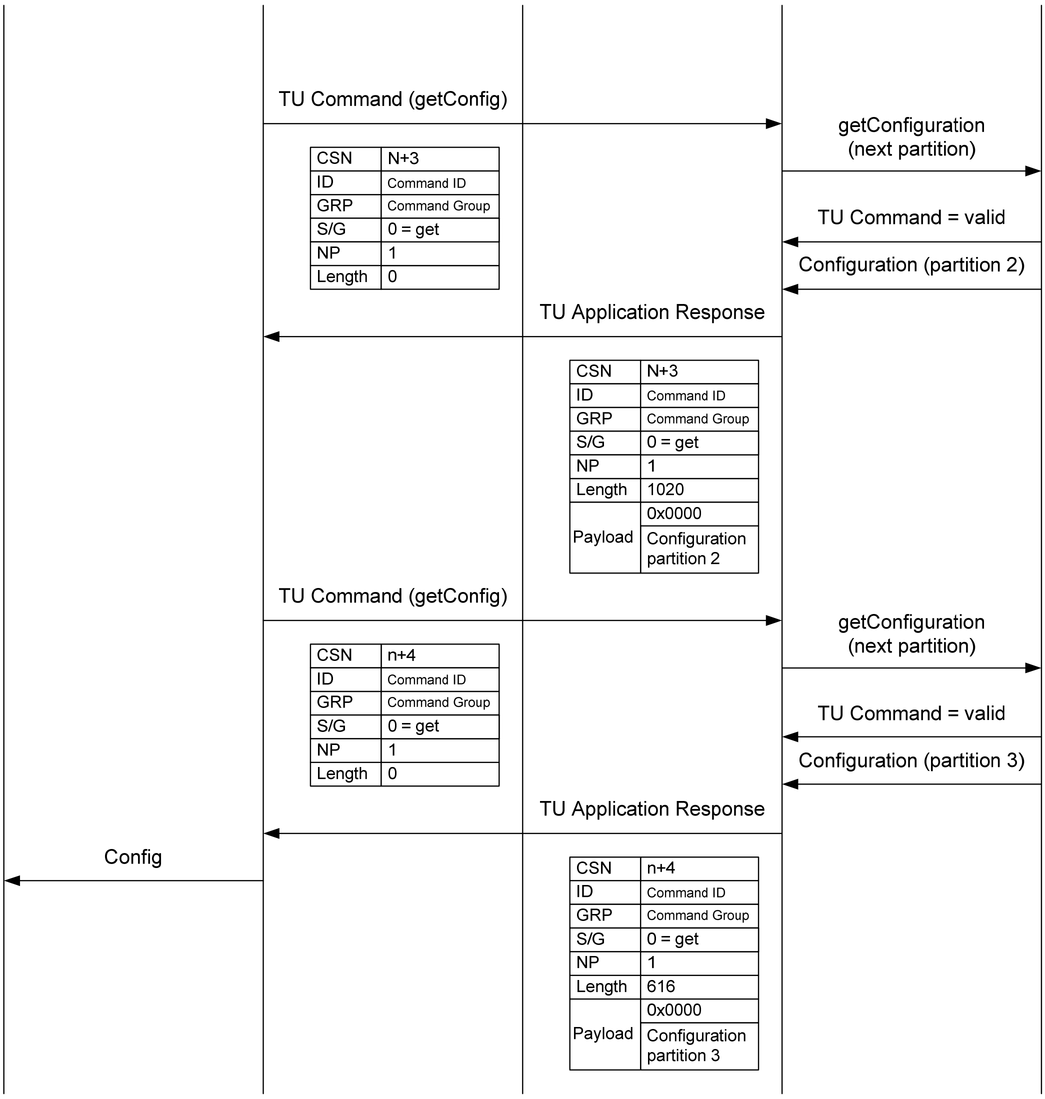


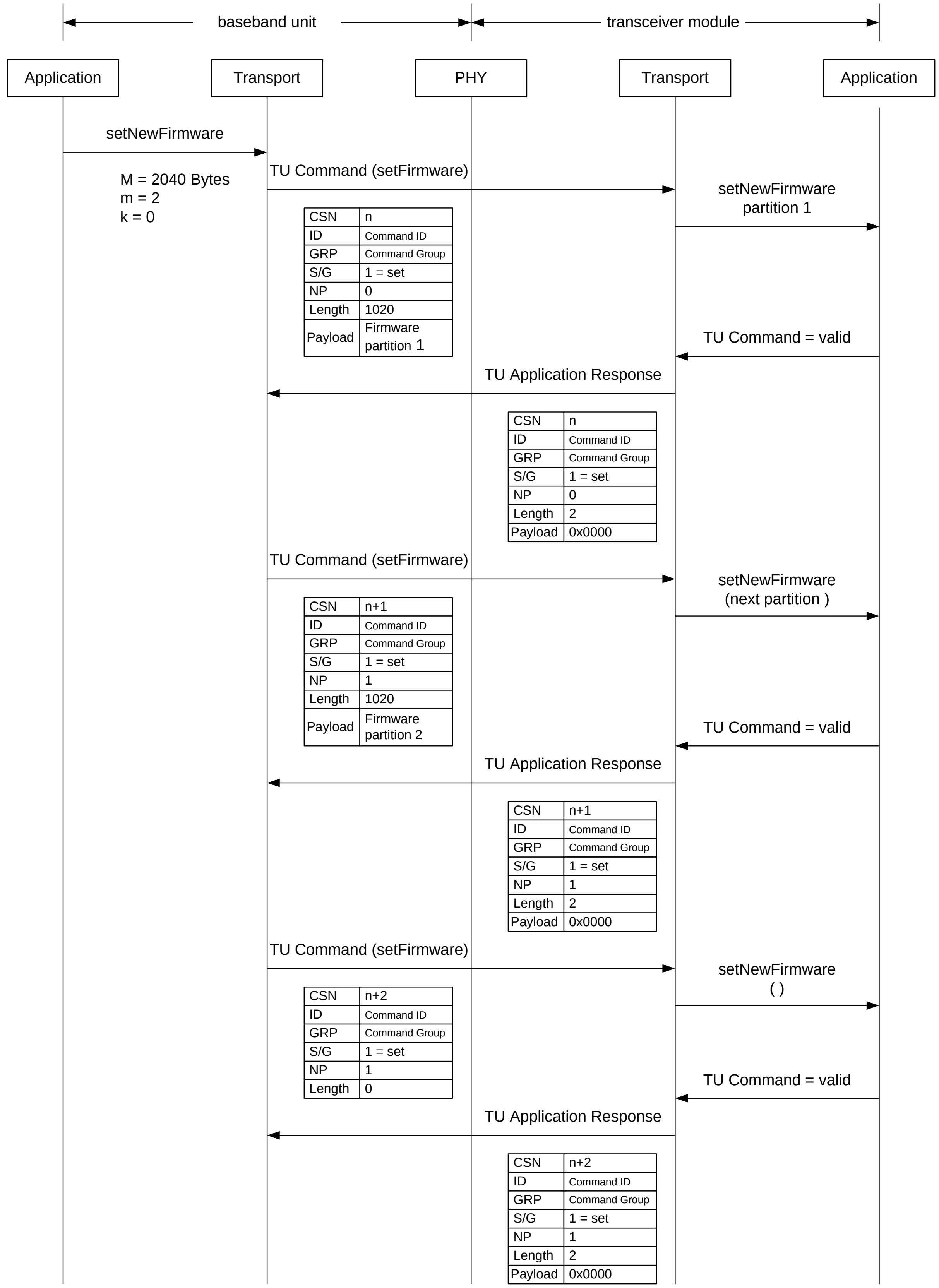
Figure 14-5: Sequence Diagram for an Interrupted and Restarted Get TU Command

14.2.4 Sequence Diagram for a Set TU Command with the Last Partition Having an Exact Length of 1020 Bytes

The BBU intends to transmit a new firmware version to the transceiver module. The data to be transmitted within the payload of the *TU command* have a length of M = 2040 bytes.

$M = 2040 = m * 1020 + k$

Hence, m = 2 and k = 0 applies.



CSN	n
ID	Command ID
GRP	Command Group
S/G	1 = set
NP	0
Length	1020
Payload	Firmware partition 1

CSN	n
ID	Command ID
GRP	Command Group
S/G	1 = set
NP	0
Length	2
Payload	0x0000

CSN	n+1
ID	Command ID
GRP	Command Group
S/G	1 = set
NP	1
Length	1020
Payload	Firmware partition 2

CSN	n+1
ID	Command ID
GRP	Command Group
S/G	1 = set
NP	1
Length	2
Payload	0x0000

CSN	n+2
ID	Command ID
GRP	Command Group
S/G	1 = set
NP	1
Length	0

CSN	n+2
ID	Command ID
GRP	Command Group
S/G	1 = set
NP	1
Length	2
Payload	0x0000

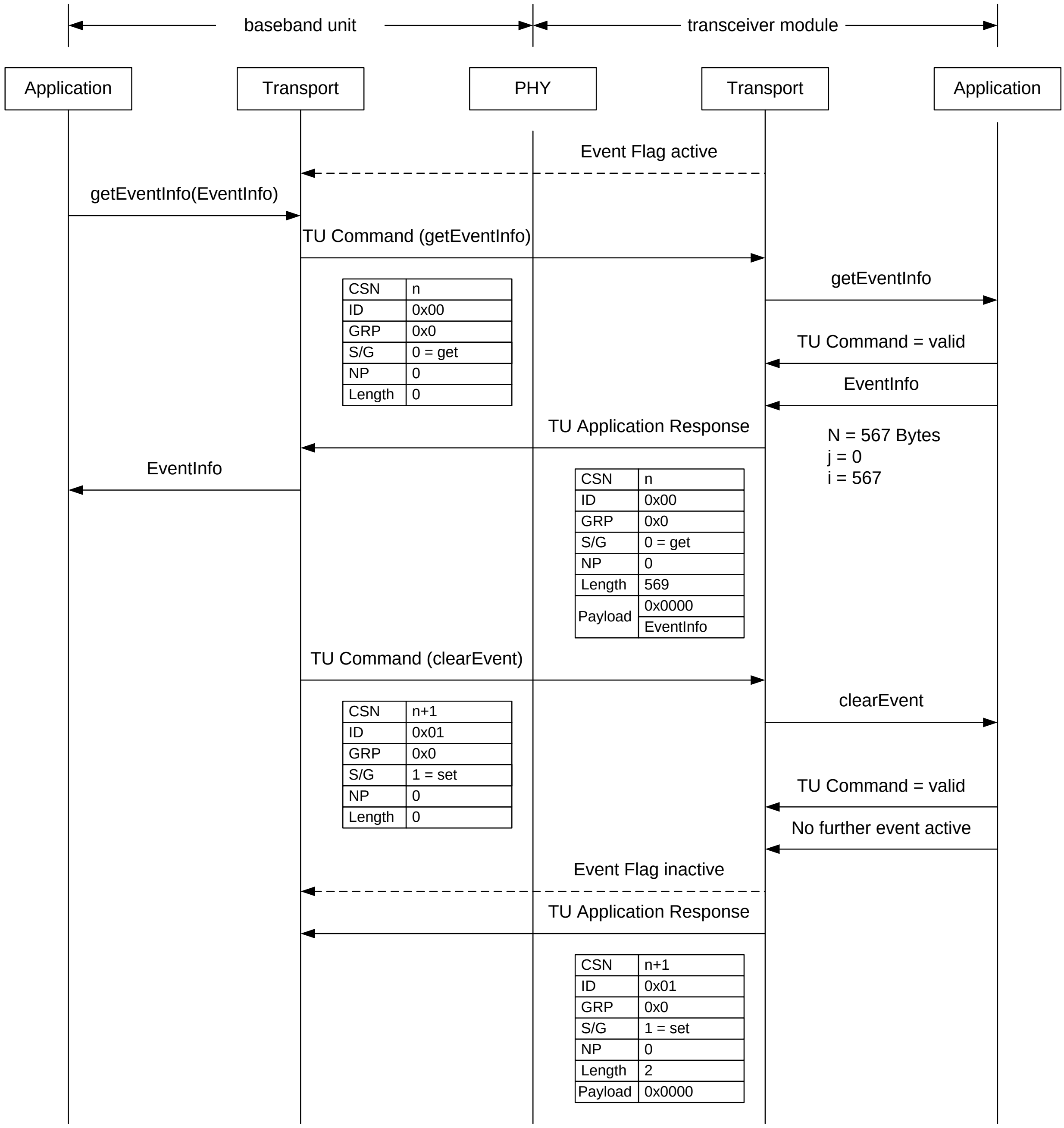
*Figure 14-6: Sequence Diagram for a Set TU Command with M = m*1020*

14.2.5 Sequence Diagram for the Case that the Transceiver Module Signals an Event

The transceiver module intends to report an event to the BBU. It sets the *event flag* in *message* n.

The EventInfo has a length of N = 567 bytes = j * 1018 + i.

Hence, j = 0 and i = 567 applies.



TU Command (getEventInfo):

CSN	n
ID	0x00
GRP	0x0
S/G	0 = get
NP	0
Length	0

TU Application Response:

CSN	n
ID	0x00
GRP	0x0
S/G	0 = get
NP	0
Length	569
Payload	0x0000 EventInfo

TU Command (clearEvent):

CSN	n+1
ID	0x01
GRP	0x0
S/G	1 = set
NP	0
Length	0

TU Application Response:

CSN	n+1
ID	0x01
GRP	0x0
S/G	1 = set
NP	0
Length	2
Payload	0x0000

Figure 14-7: Sequence Diagram for an Event Request

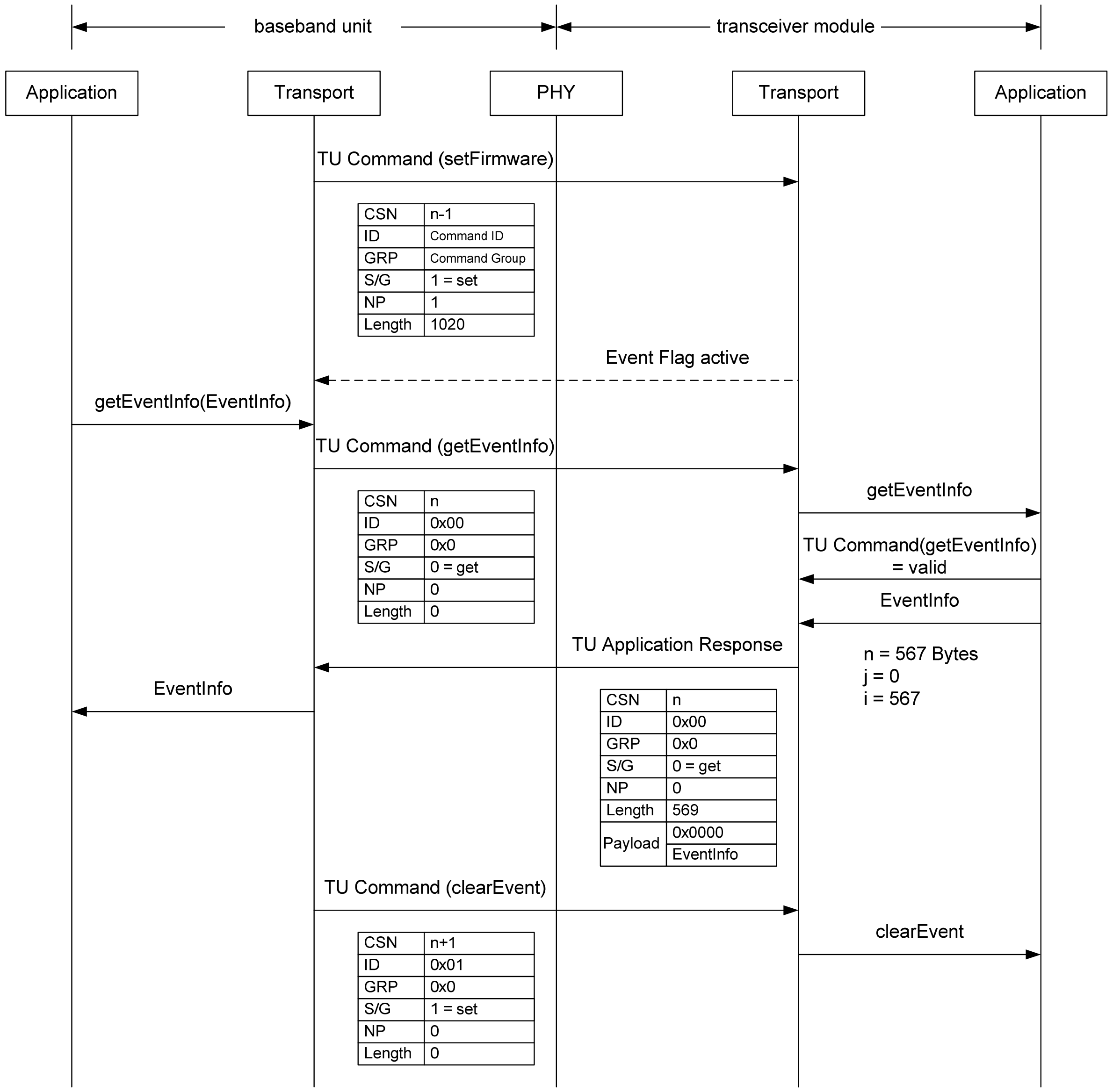
14.2.6 Sequence Diagram for the Case that the Transceiver Module Signals an Event while the BBU Is Transmitting a Larger File

The transceiver module intends to report an event to the BBU. It sets the *event flag* in *message* n. The BBU is transmitting a larger file to the transceiver module.

The EventInfo has a length of N = 567 bytes = j * 1020 + i.

Hence, j = 0 and i = 567 applies.

TU commands can be transmitted although the *TU application response* of a previous *TU command* has not yet been received. This is illustrated below as an example by transmitting the *TU application response* of the original file transfer after the ClearEvent *TU command*. The commands and responses to the file transfer are only illustrated to the extent required for illustrating the use of *event flag*.



TU Command (setFirmware)

CSN	n-1
ID	Command ID
GRP	Command Group
S/G	1 = set
NP	1
Length	1020

TU Command (getEventInfo)

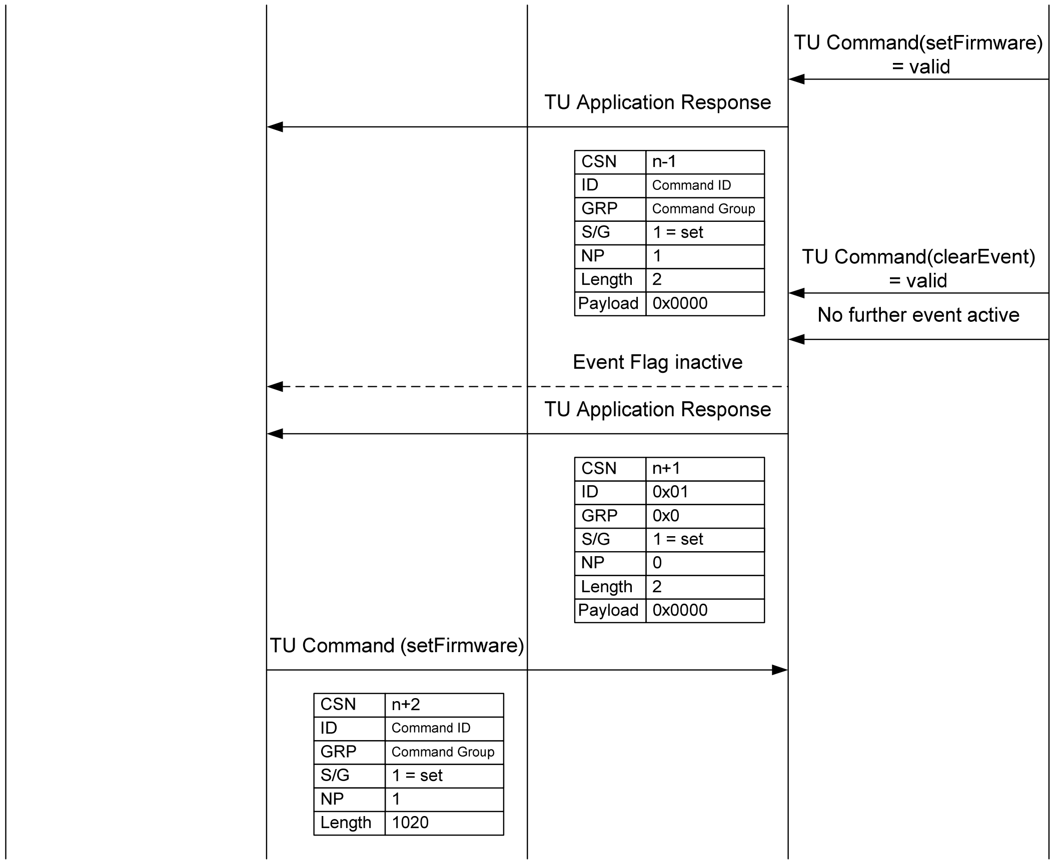
CSN	n
ID	0x00
GRP	0x0
S/G	0 = get
NP	0
Length	0

TU Application Response

CSN	n
ID	0x00
GRP	0x0
S/G	0 = get
NP	0
Length	569
Payload	0x0000 EventInfo

TU Command (clearEvent)

CSN	n+1
ID	0x01
GRP	0x0
S/G	1 = set
NP	0
Length	0



CSN	n-1
ID	Command ID
GRP	Command Group
S/G	1 = set
NP	1
Length	2
Payload	0x0000

CSN	n+1
ID	0x01
GRP	0x0
S/G	1 = set
NP	0
Length	2
Payload	0x0000

CSN	n+2
ID	Command ID
GRP	Command Group
S/G	1 = set
NP	1
Length	1020

Figure 14-8: Sequence Diagram for an Event Retrieval during a Data Transmission

14.3 Time Synchronization

Figure 14-9 illustrates the temporal references between TX and RX samples as well as the reference of an action to a TX sample and/or an RX sample.

Based on the first bit of *TX message* n, TX latency is determined by means of D_{TxL} to time of transmission $T_{TxFS,n}$ of the first sample. Time of receipt $T_{RxFS,n}$ of the first sample of *RX message* n can be determined thereof by means of D_{RxL}.

The command for action A is transmitted within *TX message* n. The TX sample reference point is specified by means of N_{BOR} and N_{SOR}. From there, the time reference of the action can be determined at the antenna connector by means of D_{TxL}. With regard to the receipt, the time reference for the action at the antenna interface can be determined by means of D_{RxL} or the RX sample reference point can be determined by means of N_{TxRx}.

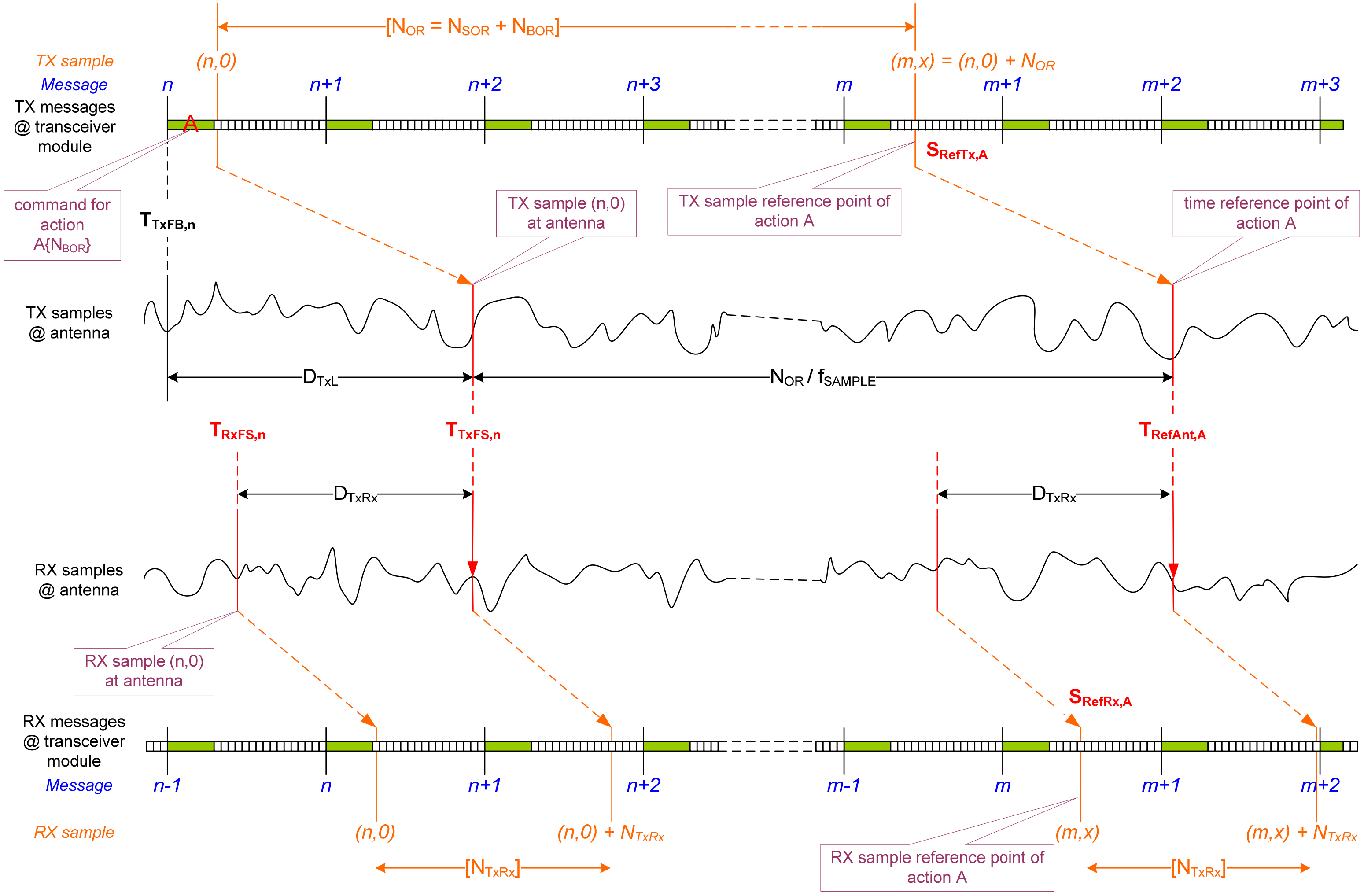


Figure 14-9: Survey of the Temporal Reference of TX Samples, RX Samples and Reference Point of an Action

14.3.1 Signal Time Synchronization of RX Samples

14.3.1.1 Implementation Variant of Signal Time Synchronization of RX Samples by means of RX Latency

Most waveforms do not impose high requirements on the temporal reference between TX and RX samples. In this case, the signal time synchronization in the receive path can be implemented on the basis of a defined transceiver module RX latency designated D_{RxL}. As illustrated in Figure 14-10, it is defined as period between time of receipt T_{RxFS} of the first sample of a *message* n at the antenna connector and time T_{RxFB} at which the first bit of *RX message* n is applied to the IQ baseband interface of the transceiver module.

$$D_{RxL} = T_{RxFB,n} - T_{RxFS,n} \qquad \text{Equation 14-1}$$

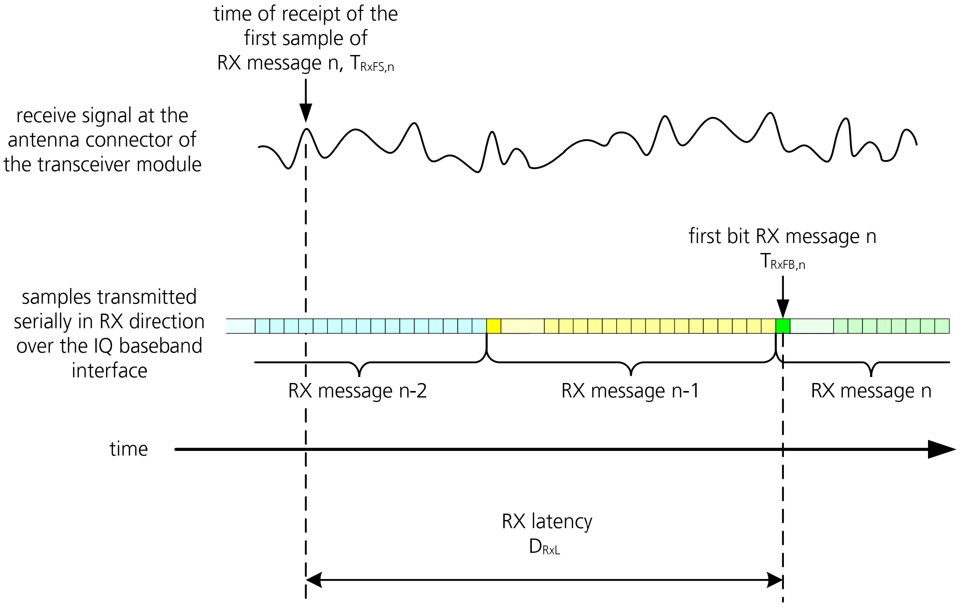


Figure 14-10: Signal Time Synchronization in the Receive Path with RX Latency Defined

Hence, the time of receipt of the first RX sample of message n at the antenna connector can, with regard to the BBU time, be calculated to

$$T_{RxFS,n} = T_{RxFB,n} - D_{RxL} \qquad \text{Equation 14-2}$$

In addition, the BBU must take the transit time over the interface from the transceiver module to the BBU into account in order to determine the time of receipt.

With regard to the BBU time, the achievable temporal accuracy for determining the time of receipt of a sample at the antenna interface depends on the temporal accuracy with which the transit time for the first bit of a *message* from the transceiver module to the input of the IQ baseband interface of the BBU can be determined and the temporal accuracy with which the BBU can detect the first bit of a *message*.

RX latency D_{RxL} can be determined from TX latency and TX/RX latency. By means of frame delay D_{TRM} between TX message and RX message, the result from Equation 8-1, Equation 8-5 and Equation 14-7 is

$$D_{RxL} = D_{TRM} + D_{TxRx} - D_{TxL} \quad \text{Equation 14-3}$$

Hence, the RX latency can be determined in the BBU from both transceiver module-specific configuration-dependent parameters D_{TxRx} and D_{TxL}.

14.3.1.2 Numerical Example for Signal Time Synchronization of RX Samples by means of RX Latency

14.3.1.2.1 Assumptions

In the transceiver module, the following transit times apply to the configuration set:

- Transit time D_{TxL} of a TX sample is 4.7 µs from the input of the IQ baseband interface of the transceiver module to the antenna interface.
- TX/RX latency is 14.9 µs.
- The transit time from feeding the samples in the serial transceiver of the IQ baseband interface of the BBU to the input of the IQ baseband interface of the transceiver module is 0.3 µs.
- The transit time from the output of the IQ baseband interface of the transceiver module to the output of samples from the serial transceiver of the IQ baseband interface of the BBU is 0.4 µs.

Furthermore, it is assumed that the waveform is working in a time slot procedure with TX sample x intended to be applied to the antenna connector exactly 50 µs after the first sample of *RX message* n has been received at the antenna connector in order to achieve the optimum temporal synchronization at the receiver.
Note: In case of a waveform, the temporal synchronization of TX and RX is not performed with regard to samples but, e.g. with regard to symbols. Within the BBU time, reference can be made to samples as a consequence.

The IQ baseband interface is operating in mode A, i.e. with 9.6 MSamples/s.

14.3.1.2.2 Results

The receive point in time of the first sample (sample 0) of *RX message* n can be determined in accordance with the implementation described in Chapter 14.3.1.1 commensurate with Equation 14-2 with the value for D_{RxL} being calculated by means of Equation 14-3 to

$$D_{RxL} = D_{TRM} + D_{TxRx} - D_{TxL} = 1.67\ \mu s + 14.9\ \mu s - 4.7\ \mu s = 11.87\ \mu s.$$

$$T_{RxFS,n} = T_{RxFB,n} - D_{RxL} = T_{RxFB,n} - 11.87\ \mu s$$

With regard to the time of receipt at the IQ baseband interface of the BBU, the time of receipt at the antenna connector can be calculated to

$$\boldsymbol{T_{RxFS,n}} = T_{RxFB,BBE,n} - 11.87\ \mu s - 0.4\ \mu s = \boldsymbol{T_{RxFB,BBE,n} - 12.27\ \mu s}$$

with the accuracy depending on how precisely the transit time from the output of the IQ baseband interface of the transceiver module to the output of samples from the serial transceiver of the IQ baseband interface of the BBU can be determined.

The waveform application intends to transmit TX sample x at time $T_{FxFS,x}$ at the antenna connector with the following having been specified by the waveform:

$$T_{TxFS,x} = T_{RxFS,n} + 50\ \mu s$$

The latency through the transceiver module is not constant since, due to *message header* and *control payload*, the samples come in asynchronously at the IQ baseband interface. The following applies to sample 0 of *TX message* m:

$T_{TxFB,m} = T_{TxFS,m} - 4.7\ \mu s.$

The BBU must feed this sample at time

$T_{TxFB,BBE,m} = T_{TxFS,m} - 4.7\ \mu s - 0.3\ \mu s = T_{TxFS,m} - 5.0\ \mu s$

in its own serial transceiver of the IQ baseband interface. For simplification reasons, it is assumed that this latency of 5 µs also applies to sample x (in practice, the BBU must compensate for this).

Thus, sample x must be fed into the serial transceiver of the IQ baseband interface of the BBU at the point in time

$\mathbf{T_{TxFB,BBE,x}} = T_{TxFS,x} - 5.0\ \mu s = T_{RxFS,n} + 45\ \mu s = T_{RxFB,BBE,n} - 12.27\ \mu s + 45\ \mu s = \mathbf{T_{RxFB,BBE,n} + 32.73\ \boldsymbol{\mu} s}.$

The accuracy of time of transmission $T_{TxFS,m}$ depends on how precisely the transit time from feeding the samples in the serial transceiver of the IQ baseband interface of the BBU to the input of the IQ baseband interface of the transceiver module can be determined and how precisely the transit time from the output of the IQ baseband interface of the transceiver module to the output of the samples from the serial transceiver of the IQ baseband interface of the BBU can be determined.

14.3.1.2.3 Additional Assumption

The assumptions of Chapter 14.3.1.2.1 continue to apply.

Now, the transceiver module is displaced for 60 m from the BBU with the additional transit time through the glass fiber cable being not made known to the BBU.

The transit time in the glass fiber cable is assumed at 5 ns/m.

14.3.1.2.4 Results

The transit time from feeding the samples in the serial transceiver of the IQ baseband interface of the BBU to the input of the IQ baseband interface of the transceiver module now is 0.3 µs + 60 * 0.005 µs = 0.6 µs.

The transit time from the output of the IQ baseband interface of the transceiver module to the output of the samples from the serial transceiver of the IQ baseband interface of the BBU now is 0.4 µs + 60 * 0.005 µs = 0.7 µs.

Hence, the correct time of receipt of sample 0 of *RX message* n would be

$\mathbf{T_{RxFS,n}} = T_{RxFB,BBE,n} - 11.87\ \mu s - 0.7\ \mu s = \mathbf{T_{RxFB,BBE,n} - 12.57\ \boldsymbol{\mu} s.}$

The BBU assumes time of receipt $T_{RxFB,BBE,n}$ – 12.27 µs, calculated above, i.e. 0.3 µs too late.

The correct point in time at which sample 0 of *TX message* m would have to be fed in the serial transceiver of the IQ baseband interface of the BBU can be calculated (taking the above simplification into account) to

$\mathbf{T_{TxFB,BBE,m}} = T_{TxFS,m} - 4.7\ \mu s - 0.6\ \mu s = T_{TxFS,m} - 5.3\ \mu s = T_{RxFS,n} + 44.7\ \mu s = T_{RxFB,BBE,n} - 12.57\ \mu s + 44.7\ \mu s = \mathbf{T_{RxFB,BBE,n} + 32.13\ \boldsymbol{\mu} s}.$

Since the BBU does not know the extended transit time over the interface, it nevertheless feeds in at time $T_{RxFB,BBE,n}$ + 32.73 µs, i.e. 0.6 µs too late.

14.3.1.3 Implementation Variant of the Signal Time Synchronization of RX Samples by means of TX/RX Latency

This example of implementation shows the way RX samples can be synchronized with TX samples independent from the accuracy of the latency of the IQ baseband interface.

14.3.1.3.1 Assumptions

The same assumptions as in the previous example apply. In the transceiver module, the following transit times apply to the set configuration:

- Transit time D_{TxL} of a TX sample is 4.7 µs from the input of the IQ baseband interface of the transceiver module to the antenna interface.
- TX/RX latency is 14.9 µs.
- The transit time from feeding the samples in the serial transceiver of the IQ baseband interface of the BBU to the input of the IQ baseband interface of the transceiver module is 0.3 µs.
- The transit time from the output of the IQ baseband interface of the transceiver module to the output of samples from the serial transceiver of the IQ baseband interface of the BBU is 0.4 µs.

Furthermore, it is assumed that the waveform is working in a time slot procedure with TX sample x intended to be applied to the antenna connector exactly 50 µs after the first sample of RX message n has been received at the antenna connector in order to achieve the optimum temporal synchronization at the receiver.

The IQ baseband interface is operating in mode A, i.e. with 9.6 MSamples/s.

14.3.1.3.2 Results

The time of receipt of the first sample of *RX message* n can be determined in accordance with Chapter 8.1.3 commensurate with Equation 8-5:

$T_{RxFS,n} = T_{TxFS,n} - D_{TxRx} = T_{TxFS,n} - 14.9\ \mu s$

In conjunction with *Equation 8-6* the following applies:

$T_{RxFS,n} = T_{TxFB,n} + D_{TxL} - D_{TxRx} = T_{TxFB,n} - 10.2\ \mu s$

With regard to the time of transmission of the first bit of *TX message* n at the IQ baseband interface of the BBU the time of receipt of the first sample of *RX message* n at the antenna connector can be calculated to

$\boldsymbol{T_{RxFS,n}} = T_{TxFB,BBE,n} - 10.2\ \mu s + 0.3\ \mu s = \boldsymbol{T_{TxFB,BBE,n} - 9.9\ \mu s}$

with the accuracy depending on how precisely the transit time from feeding the samples in the serial transceiver of the IQ baseband interface of the BBU to the input of the IQ baseband interface of the transceiver module can be determined. In this case, using an additional glass fiber cable would also result in an incorrect calculation of the time of receipt if the additional transit time is not compensated for.

If the waveform application intends to transmit the first sample of *TX message* m exactly 50 µs after receipt of the first sample of *RX message* n at the antenna connector, this sample must be applied to the IQ baseband interface exactly at

$$\boldsymbol{T_{TxFS,x}} = T_{RxFS,n} + 50\ \mu s = T_{TxFS,n} - 14.9\ \mu s + 50\ \mu s = \boldsymbol{T_{TxFS,n} + 35.1\ \mu s}.$$

In case of 9.6 MSamples/s, 35.1 µs correspond to a number of 336.96 samples. This means, the pertinent sample x must be transmitted 337 samples after sample 0 of *TX message* n, i.e. it corresponds to sample 1 of *TX message* n+21.

Hence, the synchronization of TX samples with RX samples is only depending on how precisely TX/RX latency in the transceiver module (depending on the configuration) has been determined. **At that, no dependency on the transit time over the IQ baseband interface is given.** Hence, the interface enables, in theory, a synchronization of TX samples with RX samples with an accuracy of plus/minus half a sample duration (mode A: +/- 52 ns, mode B: +/- 26 ns, mode C: +/- 10 ns, mode D: +/- 7 ns).

14.3.1.4 Transit Time Measurement Implementation Example

By means of the implementation of the signal time synchronization for RX samples described in Chapter 14.3.1.3, a highly precise transit time measurement between two radio sets (master and slave) can be performed.

The master device starts a transmit burst A at a defined time, e.g. as first sample of *message* y at time $T_{TxFS,y}$. The exact time of transmission is afflicted with the uncertainty of the latency over the IQ baseband interface. A slave device receives this burst and can, in turn, transmit a burst B with a defined delay D_{delay} (e.g. 50 µs) as described in Chapter 14.3.1.3. Delay D_{delay} and, hence, the time of transmission of this burst are independent from the transit time over the IQ baseband interface in the slave device and are only dependent on the accuracy of D_{TxRx}. The master device can associate the time of receipt of the beginning of burst B with an RX sample (e.g. sample a of *RX message* m). Since an RX sample is in very exact temporal relation to a sample transmitted, the period from the original transmit burst A to the receipt of burst B in the master device can be determined with high accuracy.

The signal transit time between master and slave device can be calculated to

$$D_{LZ} = (T_{RxFS,m} + a \,/\, f_{samp} - T_{TxFS,y} - D_{delay}) \,/\, 2. \qquad \text{Equation 14-4}$$

f_{samp} has been defined as sample rate of the interface.

Due to the fixed temporal relationship between TX and RX sample in the master device, the following applies in conjunction with Equation 8-5:

$$\begin{aligned} D_{LZ} &= (T_{TxFS,m} - D_{TxRx} + a \,/\, f_{samp} - T_{TxFS,y} - D_{delay}) \,/\, 2 = \\ &= (n \,/\, f_{samp} - D_{TxRx} - D_{delay}) \,/\, 2\,, \end{aligned} \qquad \text{Equation 14-5}$$

with n representing the number of samples between the first sample of *TX message* y and sample a of *TX message* m.

Hence, signal transit time D_{LZ} between master and slave device is independent from the transit times over the interface and only dependent on how precisely D_{TxRx} can be determined in the master (enters into Equation 14-5 by means of value D_{TxRx}) and slave (enters into Equation 14-5 by means of value D_{delay}) devices.

14.3.1.5 Example of Determining TX and TX/RX Latencies of the Transceiver Module

Determining TX/RX latency is impeded if the transceiver module is not working in full duplex since then it is more difficult to immediately relate samples transmitted to samples received.

The measurement procedure described hereinafter is based on a fixed time reference used for two consecutive measurements. For this purpose, TX latency D_{TxL} is determined in an initial measurement. By means of the TX latency known from this first measurement, another measurement serves to calculate TX/RX latency.

TX Latency Measurement:

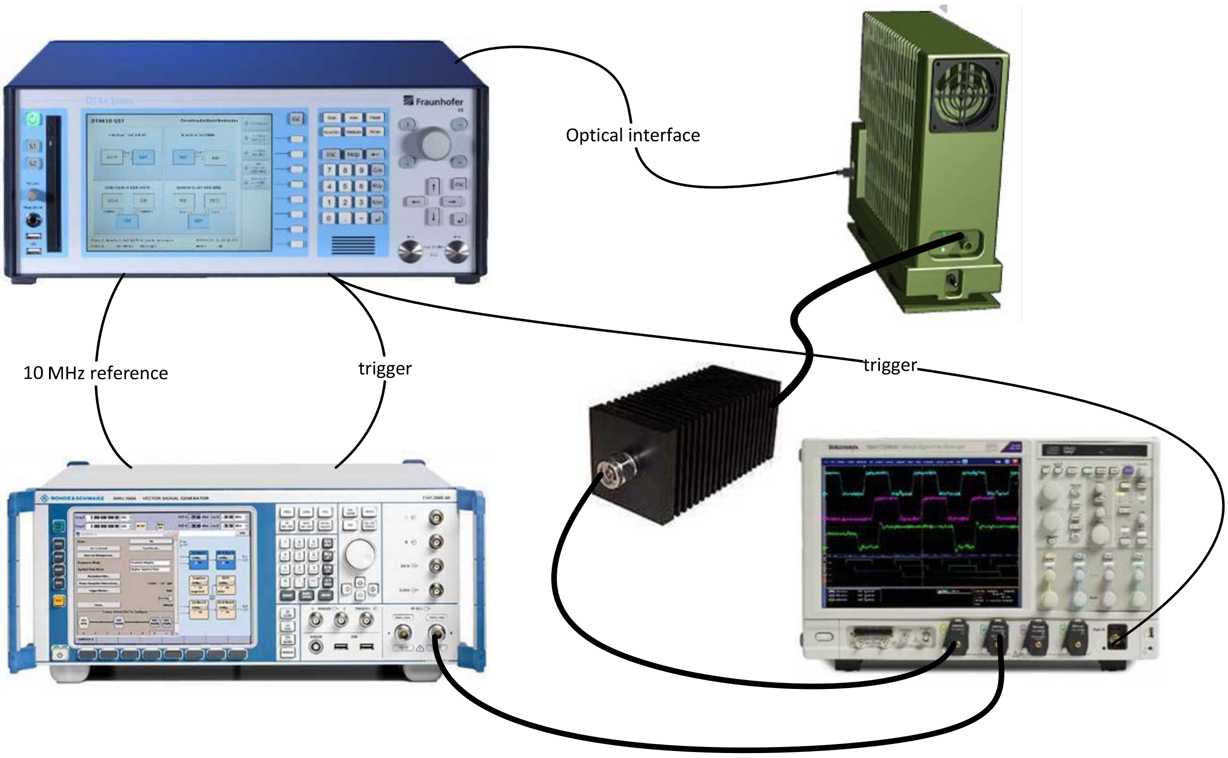


Figure 14-11: Measurement Setup for Determining TX Latency

The example in Figure 14-11 illustrates a measurement setup for determining TX latency by means of the DT4610 GST device interface test equipment specifically designed as OBISS protocol tester. This test equipment is connected via the optical interface to the transceiver module to be tested and synchronized with it. Initially, the transceiver module is in "Transmit" configuration and an appropriate transmit frequency is set. A signal generator including arbitrary waveform generator (AWG) has been synchronized with the GST via the 10 MHz reference signal; the transmit frequency set is identical to the frequency of the transceiver module. The HF outputs of transceiver module and signal generator are each connected to the input of a sufficiently fast oscilloscope; the transceiver module is connected to the oscilloscope by means of an attenuator pad to protect the input of the oscilloscope from high levels, if required.

An identical short burst signal (with a length of a few µs) has been stored in the GST and AWG. The burst signal must be conditioned in such a way that, also after filtering in the transceiver module and in the signal generator, the temporal reference between both signals can be easily determined at the oscilloscope. The signal of the signal generator should be extended with zero symbols to an exact length of one second. By doing so, the short burst is repeated every second and practically free of jitters when being transmitted periodically.

Initially, the GST is configured in such a way that the burst signal is transmitted at the antenna connector starting at the optical interface on a repetitive basis, e.g. always with the first bit of the *data payload* of *message 0* and/or at time $T_{TxFS,0}$. Furthermore, the GST shall transmit a trigger signal along with the first bit of *message 0* (SOM). This signal serves as reference point in time for the oscilloscope. Furthermore, the signal generator is configured in such a way that it modulates the one-second signal (including short bust at the beginning) on the HF carrier and also transmits it on a repetitive basis initiated by **one** trigger event. For the signal generator, the absolute reference to the trigger is not essential. The relative reference, however, to the individual bursts should be as jitter-free as possible.

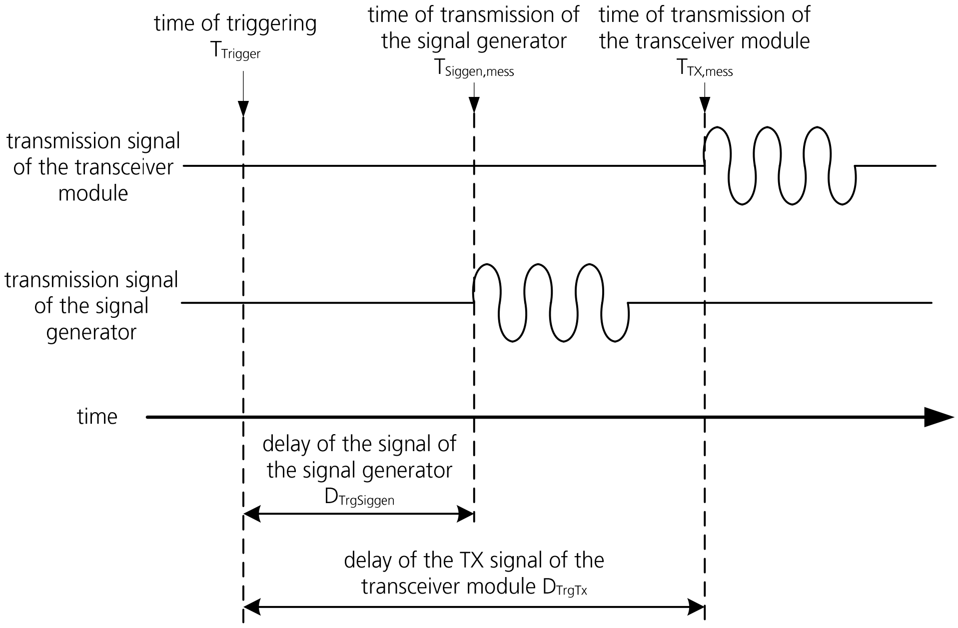


Figure 14-12: TX Latency Determination

Now, TX latency can be determined at the oscilloscope as time difference between the TX signal of the transceiver module and the trigger signal, and in addition a time difference between the TX signal of the transceiver module and the output signal of the signal generator. To increase the accuracy for TX latency, the transit times in the cables used can be taken into account in calculations.

Time $T_{TxFS,0}$ at which the first sample of *message 0* is applied to the antenna connector is calculated from the measurement values of Figure 14-12 and the transit time through the HF cable (incl. attenuator pad) $D_{RF,SEM}$ to:

$$T_{TxFS,0} = T_{Tx,mess} - D_{RF,SEM}$$

TX latency can be calculated from Equation 8-1 as follows:

$$D_{TxL} = T_{TxFS,0} - T_{TxFB,n} = T_{Tx,mess} - D_{RF,SEM} - (T_{Trigger} - D_{OptL} + D_{TriggerL}) =$$

$$D_{TrgTx} - D_{RF,SEM} - D_{OptL} + D_{TriggerL},$$

with D_{OptL} representing the transit time through the optical cable and $D_{TriggerL}$ representing the transit time through the trigger cable. If cable transit times can be neglected, TX latency can be immediately read out at the oscilloscope by means of D_{TrgTx}.

In addition, delay $D_{TrgSiggen}$ of the transmit burst of the signal generator with regard to the trigger signal is read out at the oscilloscope and written down. This value will later be used for determining TX/RX latency.

Measurement for Determining TX/RX Latency:

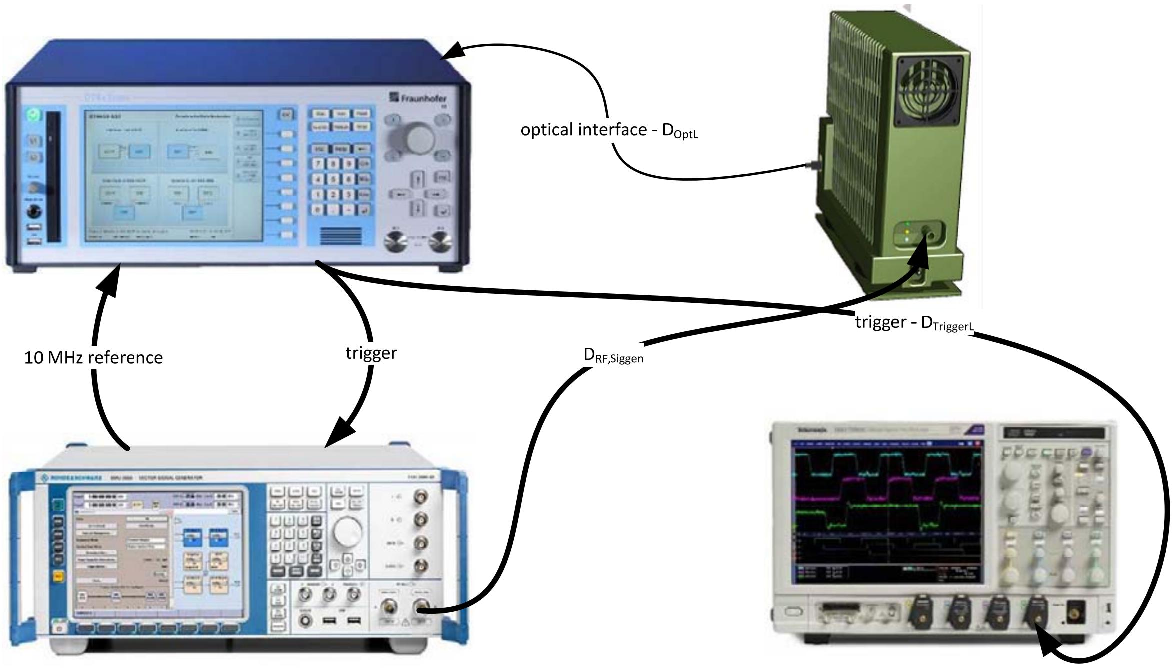


Figure 14-13: Measurement Setup for Determining TX/RX Latency

For measuring TX/RX latency, the transceiver module is now set into "Receive" state. The transmit cable of the transceiver module is disconnected and the HF cable of the signal generator is disconnected from the oscilloscope and re-connected to the transceiver module (see Figure 14-13). It is essential that the GST remains in synchronized operation with the transceiver module and the signal generator in order to not lose the temporal reference. For this purpose, the signal generator must not be stopped and the 10 MHz reference cable between GST and signal generator must not be disconnected.

Now, a snapshot of the bits received is produced in the GST, the starting time of the transmit burst received from the signal generator is searched for, and the number of samples from the first sample of RX message 0 to the first sample of the burst received is determined. Depending on the set transmission mode A to D (see Chapter 7.2.1), a period $D_{RxFS,0}$ can be calculated from the number of samples.

The time when the burst is applied to the antenna connector has been determined in the first measurement with the oscilloscope ($T_{Siggen,mess}$). By means of this value, the (virtual) point in time for the first sample of *message 0* can be calculated to:

$$T_{RXFS,0} = T_{Siggen,mess} - D_{RxFS,0}$$

$$T_{Siggen,mess} = T_{Trigger} + D_{TrgSiggen}$$

In accordance with *Equation 8-4*, TX/RX latency can be calculated accordingly:

$$D_{TXRX} = T_{TxFS,0} - T_{RxFS,0} = T_{Tx,mess} - D_{RF,SEM} - (T_{Trigger} + D_{TrgSiggen} - D_{RxFS,0}) =$$

$$= T_{Trigger} + D_{TrgTX} - D_{RF,SEM} - T_{Trigger} - D_{TrgSiggen} + D_{RxFS,0} =$$

$$= D_{TrgTX} - D_{RF,SEM} - D_{TrgSiggen} + D_{RxFS,0}$$

If the transit time through the HF cable can be neglected, TX/RX latency can be calculated to

$$D_{TXRX} = D_{TrgTX} - D_{TrgSiggen} + D_{RxFS,0}\,,$$

with D_{TrgTX} and $D_{TrgSiggen}$ having been measured at the oscilloscope and $D_{RxFS,0}$ having been calculated from the RX samples recorded.

14.3.2 Action Time Synchronization

Single actions, such as frequency hopping or amplification switching of the AGC, can be associated with an exact time in the transceiver module by means of *TC commands*. In the following, the use of the action time synchronization within a *TC command* shall be illustrated by means of specific examples.

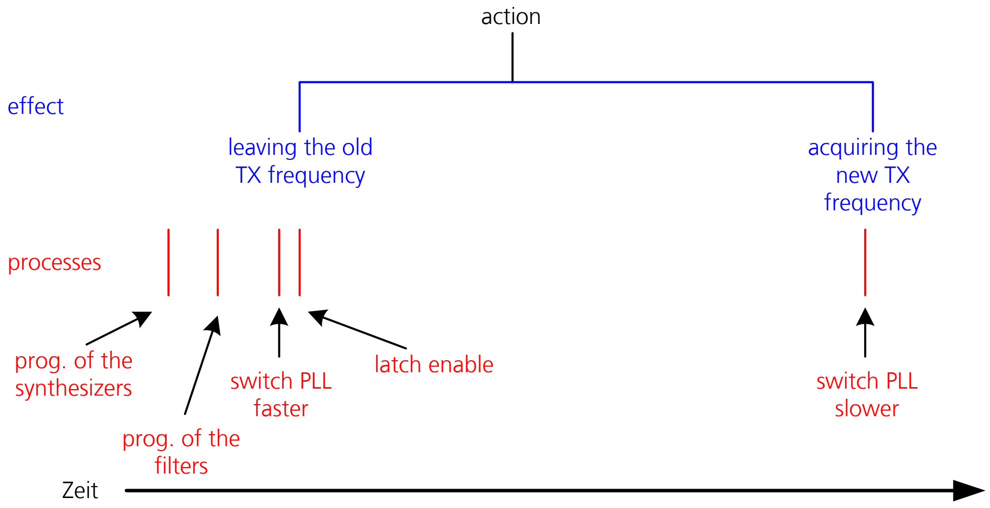


Figure 14-14: Processes and Effects of an Action (Example)

The temporal relationship of the effect(s) of an action is specified when defining the commands. The single steps performed by the transceiver module to conduct the action are called processes.

14.3.2.1 Example of Frequency Change in Transmit Mode

The transceiver module is in TX mode for the time being. The transmit frequency of the transceiver module shall be changed by means of a setTxFrequency command at a defined time.

A setTxFrequency command is transmitted within the *control payload* of a *TX message* n within a *TC command*. This is composed of

- an action identification (here: setTxFrequency, signaled by the *TC command ID*)
- an NBOR time delay in the form of a number of samples (N_{BOR})
- additional information with regard to the action within the *TC command payload* (here: new frequency)

Hence, **one** temporal reference point can be signaled with each *TC command*. The command must be defined in such a way that the temporal relationship of the effect(s) of the action can be derived from this.

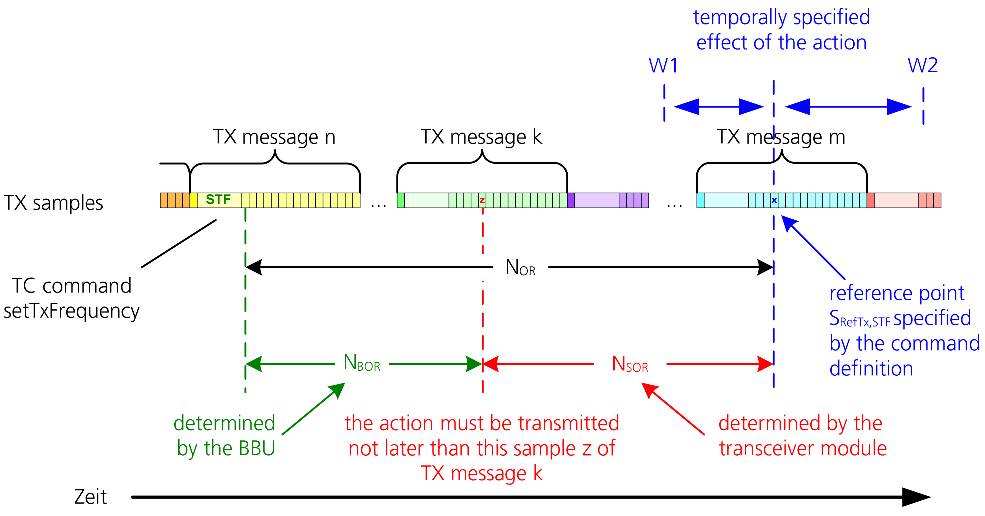


Figure 14-15: Signaling of the Reference Point of the setTxFrequency Action

Figure 14-15 illustrates the temporal reference of signaling the setTxFrequency command within the *TC command* and of the sequence of processes of an action in the transceiver module to reference point S_{RefTx}. This temporal reference is specified by

- the command definition: it specifies the definition of the TX sample reference point with regard to the sequence of the action(s) (in blue color)
- the transceiver module: it specifies the number of samples that must be transmitted not later than the TX sample reference point of the *TC command* (in red color)
- the BBU: it specifies the message in which the *TC command* is transmitted (in green color)

14.3.2.1.1 Command Definition

By means of the setTxFrequency command, an action is performed in the transceiver module. If required, the action can be split into various processes (synthesizer programming, switching of filters etc.) that must be performed at a prescribed time. Such processes follow a fixed temporal reference to each other with regard to the reference point of the action. The sequence of the processes is specified by the design of the transceiver module and they take place in a fixed temporal reference to each other. By means of these processes, effects are achieved that, in turn, are in a fixed temporal reference to each other and, hence, also to the reference point.

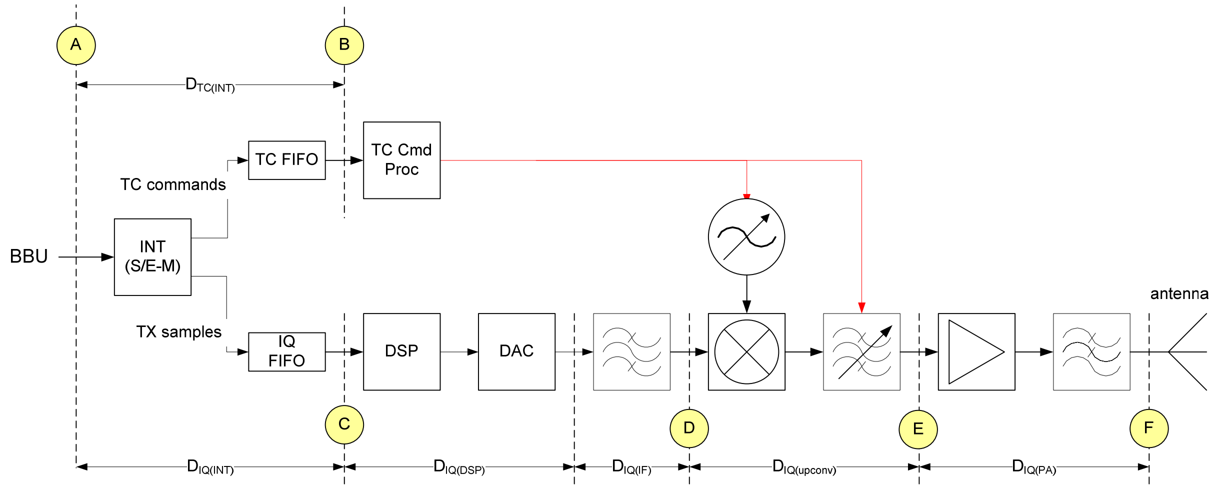


Figure 14-16: Example of a Simplified Implementation of a Transceiver Module

Figure 14-16 illustrates the example of a simplified block circuit diagram of the transmit portion of a transceiver module. Point A represents the IQ baseband interface at the input of the transceiver module. Within the first INT function block, the incoming serial data stream is split in *TC commands* (here, the handling of *TU commands* is not essential and will not be taken into account) and in TX samples. These are each stored in FIFO memories for further processing.

TX samples are fed through a number of blocks not affected by frequency hopping in this example (DSP, DAC, IF filter as well as PA and harmonic filter) and blocks affected by frequency hopping (up-converter and RF filter).

The switching process is performed within a finite period of time within which the blocks affected are in an undefined state. So that during this period no uncontrolled transmissions are performed, the transmit power must initially be reduced to a minimum in a controlled manner. In general, this is conducted by means of the complex-valued TX sample amplitude that is reduced to "0" within a specified time interval. As long as the transmit signal has not yet been sufficiently faded away the transceiver module state responsible for the transmission of the TX samples must not change. However, initial processes preparing the change of frequency may be performed (e.g. programming the new frequency in the synthesizer). As soon as the last sample valid for original frequency f_{old} has passed point E, the switching process may commence. This applies accordingly to additional hardware components (e.g. ATU) possibly following after the transceiver module.

After the switching process, the performance of new frequency f_{new} is increased again at the transceiver module. This can also be conducted by means of the performance in the TX samples. Synthesizer and RF filter must be switched to the new frequency (i.e. must be present within a tolerable frequency range around new set frequency f_{new}) before the first sample to be transmitted with new frequency f_{new} set comes in at point D.

Thus, both points in time—when the last sample (with f_{old} set) passes point E and when the first sample (with f_{new} set) comes in at point D—are essential for signaling frequency hopping. These can be expressed by means of a hold time and a setup time. This is illustrated in Figure 14-17.

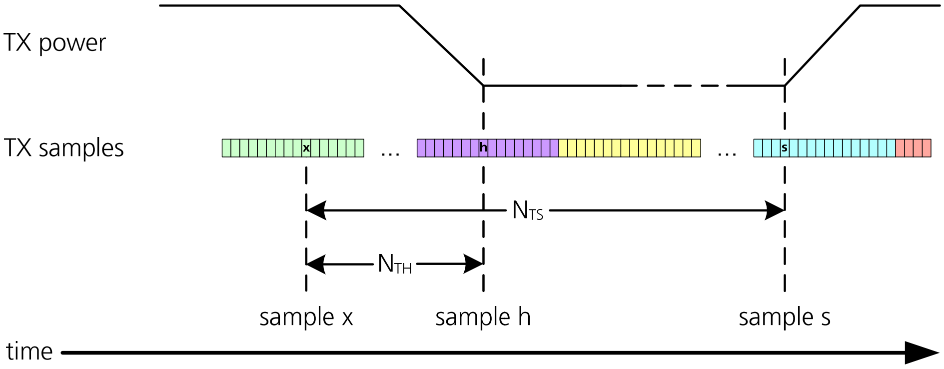


Figure 14-17: Setup and Hold Time at the setTxFrequency Command

TX sample x has been signaled as TX sample reference point for the action.

Hold time is the time that is required by last sample h to pass point E with frequency f_{old} set. By that point in time, the transmit power must have been reduced. In the following, sample h is marked by the number of N_{TH} samples starting from TX sample reference point x.

Setup time is the time required by the transceiver module until the frequency change has been completed. The end is marked by the point in time when first sample s transmitted with new frequency f_{new} set comes in at point D. At this time, the transmit power can be increased again. Sample s is marked by the number of N_{TS} samples starting from the TX sample reference point.

For this purpose, the relationship to TX sample reference point x may be defined at random in the *TC command*. Hence, N_{TS} or N_{TH} can have positive (in case sample h or s come after sample x) or negative (in case sample x comes after samples h or s) values or have the value "0" in case the TX sample reference point is equated with one of samples h or s.

In this example, it is assumed that the following has been defined in the setTxFrequency command definition:

The TX sample reference point corresponds to the sample that shall be transmitted first with new frequency f_{new} set.

Hence, N_{TS} is specified to "0", N_{TH} is a negative value (see Figure 14-18). This must be made known to the BBU before it can use that command.

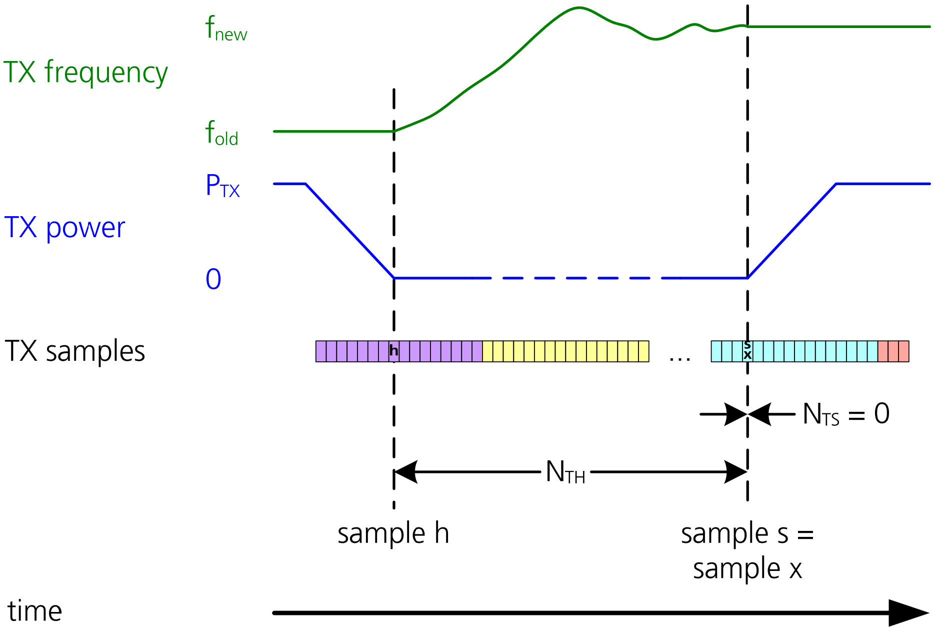


Figure 14-18: TC Command with the Definition Setup Time = 0

How can N_{TH} be determined?

Hence, the following applies: The effect of the action is complete (i.e. the transceiver module can transmit with new frequency f_{new}) when the TX sample reference point has reached the antenna connector (= T_F). Synthesizer and filter must be set to new frequency f_{new} by the time this sample reaches point D (T_D). With regard to the antenna interface the following can be calculated

$$T_D = T_F - D_{IQ(PA)} - D_{IQ(upconv)} \qquad \text{Equation 14-6}$$

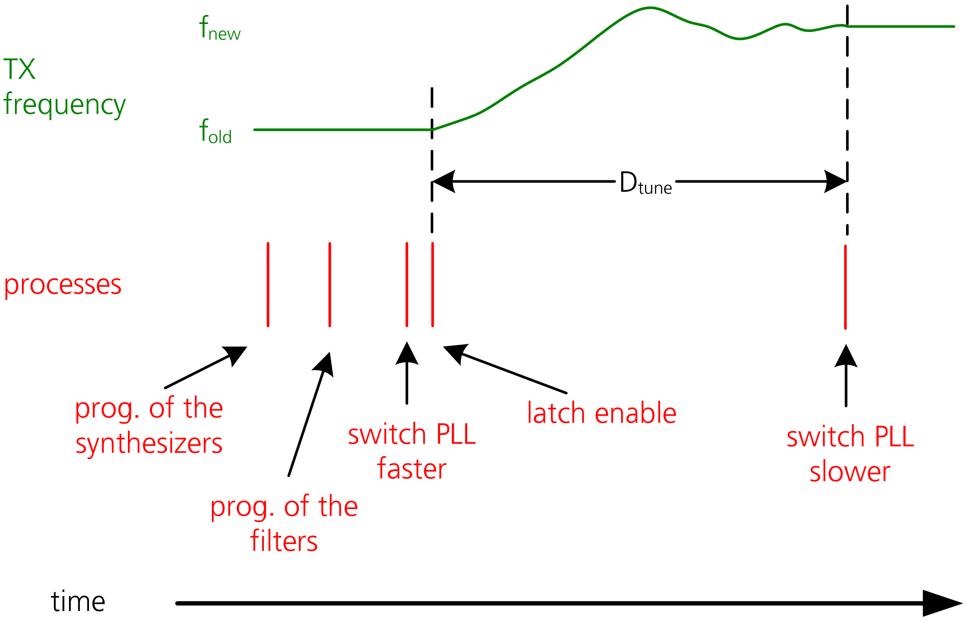


Figure 14-19: TC Command with the Definition Setup Time = 0

The effect of the action must not begin earlier than the time the last sample to be transmitted with frequency f_{old} set has reached point E. This means that at this time, the *TC command* processor must start the tuning process. This would, for example, correspond to the "Latch Enable" command to the synthesizer by means of which the frequency programmed is transferred to the synthesizer. Processes such as programming the new frequency in the synthesizer can be performed in advance if they have no impact on the HF behavior of the synthesizer.

It is assumed that period D_{tune} is required to tune synthesizer and filter to the new frequency.

If it is assumed that N_{TH} is predefined (i.e. the number of samples between last sample h with f_{old} set and first sample s with f_{new} set is predefined by the waveform), then D_{tune} can be calculated by means of sample frequency f_{samp} to:

$$D_{tune} = N_{TH} / f_{samp} - D_{IQ(upconv)} \qquad \text{Equation 14-7}$$

With D_{tune} predefined by the hardware the following can be concluded for N_{TH}:

$$N_{TH} = ceil\,[(D_{tune} + D_{IQ(upconv)}) * f_{samp}] \qquad \text{Equation 14-8}$$

14.3.2.1.2 Definition of N_{SOR}

Each transceiver module must determine a minimum number of TX samples N_{SOR} by which a command must be transmitted prior to the reference point so that it can be performed in good time under all conditions (if required, for frequency hopping several values can be defined for N_{SOR}, e.g. depending on the frequency range). Based on TX sample reference point $S_{RefTx,STF}$ (= sample x of *message* m = the first sample to be transmitted with new frequency f_{new} set), N_{SOR} indicates to sample z of *message* k. The command must be transmitted not later than this sample so that it can be performed in due time in the transceiver module (see Figure 14-15). N_{SOR} is specific for the transceiver module and possibly depending on the configuration and must be made known to the BBU before the setTxFrequency command can be used.

How can N_{SOR} be determined for the transceiver module?

The tuning process of synthesizer and filter with duration D_{tune} must be started not later than T_{start}

$$T_{start} = T_F - D_{IQ(PA)} - D_{IQ(upconv)} - D_{tune} \qquad \text{Equation 14-9}$$

with T_F corresponding to the time when the TX sample reference point reaches the antenna interface (it is taken into account for this that the process must already be completed when the TX sample reference point reaches point D). If an additional time $T_{pretune}$ is considered for processes taking place before the actual tuning process, the command must come in at the IQ baseband interface of the transceiver module (point 1) not later than

$$T_{start} = T_F - D_{IQ(PA)} - D_{IQ(upconv)} - D_{tune} - D_{pretune} - D_{TC(INT)} \qquad \text{Equation 14-10}$$

The time difference to when the command with regard to the TX sample reference point must be transmitted over the interface can be calculated to

$$D_{SOR} = D_{IQ(PA)} + D_{IQ(upconv)} + D_{tune} + D_{pretune} + D_{TC(INT)} \qquad \text{Equation 14-11}$$

By means of sample frequency f_{samp} the following results from Equation 14-11:

$$N_{SOR} = ceil\ [(D_{IQ(PA)} + D_{IQ(upconv)} + D_{tune} + D_{pretune} + D_{TC(INT)}) * f_{samp}] \quad \text{Equation 14-12}$$

14.3.2.1.3 Definition of N_{BOR}

N_{SOR} is a value specific for the transceiver module and possibly depending on the configuration that must be made known to the BBU for one time and is not transmitted within the *TC command*. By means of the N_{SOR} parameter it is specified that the command must be transmitted not later than this sample. The BBU specifies when the command is actually transmitted to the transceiver module. N_{BOR} indicates the offset between sample z and the first sample of the message in which the command is transmitted. In accordance with Figure 14-15 the N_{BOR} value can be calculated to

$$N_{BOR} = z + (k - n) * N_{Message} \quad \text{Equation 14-13}$$

For this purpose, $N_{Message}$ is the number of samples within a *message* for the mode used; k is the number (MC) of the *message* with the sample at which the command must be transmitted at the latest and n is the number of the *message* in which the command is transmitted.

	$N_{Message}$
Mode A	16
Mode B	32
Mode C	80
Mode D	120

Table 14-1: Number of Samples per Message, Depending on the Mode Used

In accordance with the specification (see *Equation 8-9*) N_{BOR} can have values between 0 and 4094. In case of $N_{BOR} = 0$, N_{BOR} indicates to the first sample (= sample 0) of the *message* in which the *TC command* is also transmitted.

The entire delay from transmitting the command to reference point $S_{RefTX,STF}$ (= first sample to be transmitted with the new frequency set) can be calculated to

$$N_{OR} = N_{SOR} + N_{BOR} = x + (m - n) * N_{Message} \quad \text{Equation 14-14}$$

For this purpose, $N_{Message}$ is the number of samples within a *message* for the mode used; m is the number of the *message* with the TX sample reference point; n is the number of the *message* in which the command is transmitted and x is the position of the TX sample reference point in *message* m.

14.3.2.2 Numerical Example

In the following, it is described by means of a specific numerical example how the example of the setTxFrequency command defined in Chapter 14.3.2.1.1 can be used. The example is explicitly related to the BBU and not to the waveform since the waveform must additionally take the delay periods occurring in the BBU into account, as appropriate. These periods are not subject to the specification of the interface between BBU and transceiver module.

14.3.2.2.1 Assumptions

The BBU intends to change the frequency in transmit mode. The TX sample reference point corresponds to sample No. 13 of *TX message* 12500. The TX sample reference point has been defined as the first sample to be transmitted with the new frequency set.

Additional assumptions are:

- Mode A is selected for transmitting the sample, i.e. the sample rate is 9.6 MSample/s.
- The D_{tune} time required to switch to a new frequency is 30 µs.
  The processes prior to the actual tuning process require a period of 6 µs.
- The individual transit times of a TX sample in the transmit path are
  $D_{IQ(INT)} = 2.5$ µs
  $D_{IQ(DSP)} = 8.4$ µs
  $D_{IQ(IF)} = 1.5$ µs
  $D_{IQ(upconv)} = 0.1$ µs
  $D_{IQ(PA)} = 0.1$ µs
- The transit time of the *TC command* to the *TC command* processor is $D_{TC(INT)} = 2.5$ µs
- The BBU will insert the *TC command* at the latest time possible.

14.3.2.2.2 Results

Hence, N_{SOR} from Equation 14-12 can be calculated to

$$\boldsymbol{N_{SOR}} = ceil\,[D_{IQ(PA)} + D_{IQ(upconv)} + D_{tune} + D_{pretune} + D_{TC(INT)}) * f_{samp}] =$$
$$= ceil\,[(0.1\,\mu s + 0.1\,\mu s + 30\,\mu s + 6\,\mu s + 2.5\,\mu s) * 9.6*10^6\,s^{-1}] =$$
$$= ceil\,(38.7\,\mu s *9.6*10^6\,s^{-1}) = ceil(371.52) = \mathbf{372}$$

The *message counter* value of the *message* in which the command must be transmitted at the latest can be calculated to

$$\boldsymbol{MC_{SOR}} = floor\,[MC_{RefTx} - (N_{SOR} - x\,) \,/\, N_{Message}] =$$
$$= floor\,[\,12500 - (372 - 13) / 16] = floor\,(12477.5625) = \mathbf{12477}$$

with MC_{RefTx} being the *message counter* value of the *message* with the TX sample reference point and x being the sample number of the TX sample reference point within the *message*.

N_{BOR} can be calculated to

$$\boldsymbol{N_{BOR}} = N_{Message} - rest[(N_{SOR} - x) \,/\, N_{Message}] =$$
$$= \{16 - rest\,[(372 - 13) / 16]\,\}\; MODULO16 = \mathbf{9}$$

In this case, the setup time is equal to "0" since the TX sample reference point has been selected in such a way that it marks the sample transmitted first with the new frequency set over the interface, i.e. when the effect of the action is complete.

Hence, $\boldsymbol{N_{TS} = 0}$ applies.

From Equation 14-8, N_{TH} can be calculated to

$$\boldsymbol{N_{TH}} = ceil\,[(D_{tune} + D_{IQ(upconv)}) * f_{samp}] =$$

$$= ceil\,[(30\ \mu s + 0.1\ \mu s) * 9.6{*}10^{6}\ s^{-1}] = ceil\,(288.96) = \boldsymbol{289}$$

$$\boldsymbol{MC_{TH}} = floor\,[MC_{RefTx} - (N_{TH} - x) / N_{Message}] =$$

$$= floor\,[12500 - (289 - 13) / 16] = floor\,(12482.75) = \boldsymbol{12482}$$

$$\boldsymbol{S_{TH}} = N_{Message} - rest[(N_{TH} - x) / N_{Message}] =$$

$$= \{16 - rest\,[(289 - 13) / 16]\,\}\ MODULO16 = \boldsymbol{12}$$

Hence, the last sample with the original frequency set is **sample 12** of the *message* with the *message counter* value **12482**.

***** End of Document *****