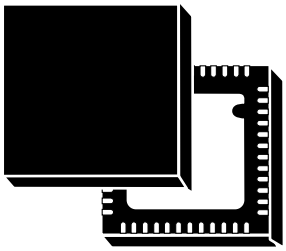
TN100

High performance CSS transceiver enabling location awareness

Preliminary Data

Features

- Single-chip solution for ISM 2.45 GHz RF transceiver
- Built-in ranging capability for link distance estimation
- Modulation technique: chirp spread spectrum (CSS)
- FDMA support: 7 channels, 3 non-overlapping
- Data rates: 2 and 1 Mbps; 500, 250 and 125 kbps
- Adjustable output power from -33 dBm to 0 dBm
- Receiver sensitivity: -97 dBm @ 250 kbps, FEC, and BER=10-3
- In-band carrier-to-interference C/I = 0 to 3 dB @ 250 kbps & C=-80 dBm
- Supports external power amplifier to increase output power
- High precision ranging: 2 m indoors and 1 m outdoors
- Supply voltage range from 2.3 V to 2.7 V
- Current consumption: 27 mA (RX), 30 mA (TX) @ 0 dBm
- Standby current with active RTC: 2 µA
- RSSI sensitivity: -95 dBm
- 32.768 kHz clock available for external MCU
- Integrated fast SPI interface (27 Mbps)
- Integrated MAC controller with FEC, CRC checking
- CSMA/CA, FDMA, and TDMA modes
- Automatic retransmission and acknowledgement
- Automatic address matching
- Industrial temperature range from -40°C to +85°C
- VFQFPN2-48 ECOPACK® package



VFQFPN2-48

Description

The TN100 transceiver is a highly integrated mixed signal chip that uses the wireless communication technology CSS (chirp spread spectrum) developed by Nanotron Technologies.

With its unique ranging capability, TN100 can measure the link distance between two nodes. Thus, TN100 supports location awareness applications including location based services (LBS) and asset tracking (2D/3D RTLS). Ranging is performed during regular data communication and does not require additional infrastructure, power, and/or bandwidth.

For an even better ranging accuracy, a high precision mode is provided. SDS-TWR algorithm (symmetrical double-sided two-way ranging) allows superior accuracy even with the use of low cost crystals for the oscillators.

Contents

1 Summary description

The TN100 transceiver IC is designed to build robust, short distance wireless networks operating in the 2.45 GHz ISM band with extremely low power consumption over a wide range of operating temperatures.

The TN100 supports 7-frequency channels with 3 non-overlapping channels. This provides support for multiple physically independent networks and improved coexistence performance with existing 2.4 GHz wireless technologies. Data rates are selectable between 31.25 kbps and 2 Mbps. Due to the chip's unique chirp pulse, adjustment of the antenna is not critical. This significantly simplifies the system's installation and maintenance (“pick-and-place”).

The TN100 transceiver includes a sophisticated MAC controller with CSMA/CA and TDMA support as well as forward error correction (FEC) and 128-bit hardware encryption. It also provides scrambling, automatic address matching, and packet retransmission, thus minimizing the requirements for microcontroller and software.

Through its high-speed standard SPI interface, the TN100 can be interfaced with a wide range of external microcontrollers. It includes a 4-Kbit frame buffer which allows even very slow microcontrollers to work with the transceiver. This means that several receive and transmit frames can be stored simultaneously in the buffers. This solution eliminates the problems of different peak data rates between air and microcontroller interfaces.

In order to meet environmental requirements, ST offers these devices in ECOPACK® packages. ECOPACK® packages are lead-free. The category of second Level Interconnect is marked on the package and on the inner box label, in compliance with JEDEC Standard JESD97. The maximum ratings related to soldering conditions are also marked on the inner box label.

ECOPACK is an ST trademark. ECOPACK specifications are available at: www.st.com.

Target applications

Target applications for the TN100 include:

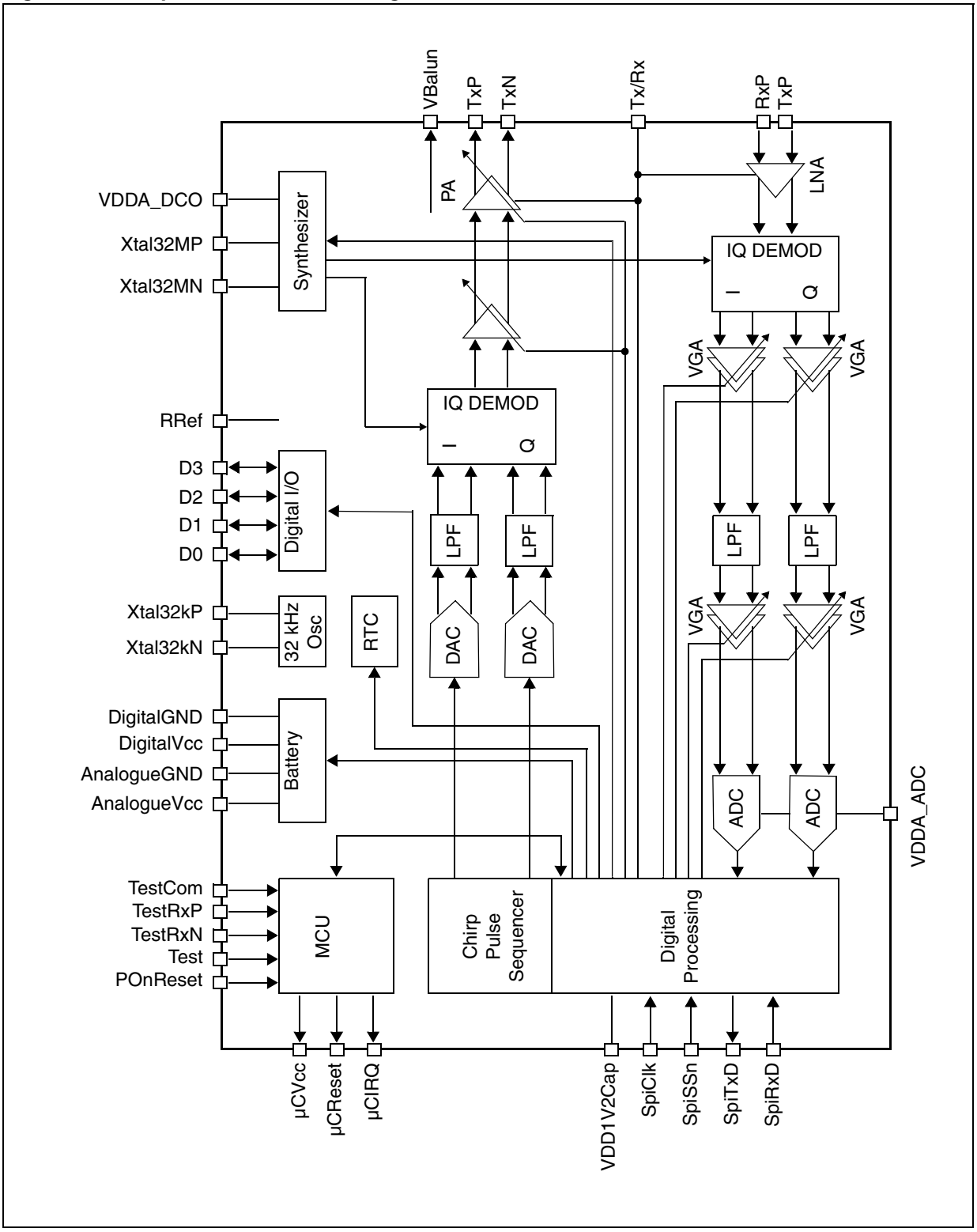
- Asset tracking
- Enabling 2D/3D real-time location systems
- Security
- Industrial monitoring and control
- Medical applications

Development environment

- Simple API access to chip registers
- Easy-to-use evaluation boards for testing the TN100 in any environment
- Ready-to-customize development boards for quick application development

2 TN100 block diagram

Figure 1. Simplified TN100 block diagram



3 Key features

The TN100 transceiver incorporates robust wireless communication capabilities including:

- Built-in precise ranging capabilities
- Channelization using FDMA for improved coexistence
- Different symbol durations and symbol rates
- Standard set of default register values set into chip
- Digital Dispersive Delay Line (DDDL) incorporated in the chip
- Programmable pull resistors

3.1 Built-in ranging capability

The TN100 transceiver provides a built-in ranging capability. The chip can be enabled to provide Round Trip of Flight (RToF) information using a special Data / Ack packet transmission. Because the processing time for generating a hardware acknowledgment is known and the time of transmission a data packet is known, these two values can be used to calculate a ranging distance between the two TN100 stations.

3.2 Channelization using FDMA for improved coexistence

Channelization is the subdividing of the available frequency band (in this case, the 83 MHz ISM band) into many narrow frequency bands, which for the TN100 transceiver is 22 MHz channels. This increases coexistence with devices sharing the same ISM band. The TN100 transceiver channelizes the 2.4 GHz ISM band into multiple 22 MHz frequency channels using Frequency Division Multiple Access (FDMA). With register settings, the chip can be set to 80 MHz or 22 MHz signal bandwidth. The chip is set by default to use FDMA on startup and can be programmed to use one of multiple 22 MHz channels in the ISM band.

3.3 Incorporated digital dispersive delay-line (DDDL)

The TN100 transceiver incorporates a Digital Dispersive Delay Line (DDDL), which in previous chips was an external component. Reference values for I and Q can be stored in Baseband RAM. These values are used to detect incoming signals generated by another transceiver chip. The detectable signals can be one of the following:

- An Upchirp (linear frequency modulation, where frequency increases in time)
- A Downchirp (linear frequency modulation, where frequency decreases in time)
- An OffChirp, which is the absence of a chirp

Note: For more details, see Section 20: Chirp modulation on page 89.

3.4 Selectable symbol durations and symbol rates

The TN100 transceiver provides selectable symbol durations and symbol rates.

- Symbol durations include: 0.5 µs, 1 µs, 2 µs, and 4 µs
- Symbol rates include: 2 Mbps, 1 Mbps, 0.5 Mbps and 0.25 Mbps

Note: See Section 26.2.61: 0x48 – Symbol duration, symbol rate, and modulation system on page 169.

4 Architecture

The TN100 is a extremely low power, highly integrated mixed signal chip incorporating both an analog and a digital part in one silicon die. This section provides a brief overview of the digital and analog parts of the chip.

4.1 Analog part - TX and RX

For transmission, the analog part of the chip converts data obtained from the digital part into chirp pulses and sends packets over the air using an RF link. For reception, it detects received chirp pulses into a form that can be used by the digital part.

4.2 Digital part - programming interface

The digital part of the chip provides an SPI interface for reading and writing to chip registers. Application data is written to chip buffers in the digital part, which is then provided to the analog part of the chip for transmission over the air. Data received from the analog part is provided to an application via buffers in the digital part. To achieve maximum power savings, the digital part of the TN100 transceiver is divided into two sections called an *ON* section (that is, always powered) and a *PWD* section (that is, powered during operating mode).

4.2.1 Digital part – ON Section

The ON section, which is always powered, contains the minimum number of required controls that are used to maintain the chip settings. It also is connected to the digital I/O pins as these pins are used for power management.

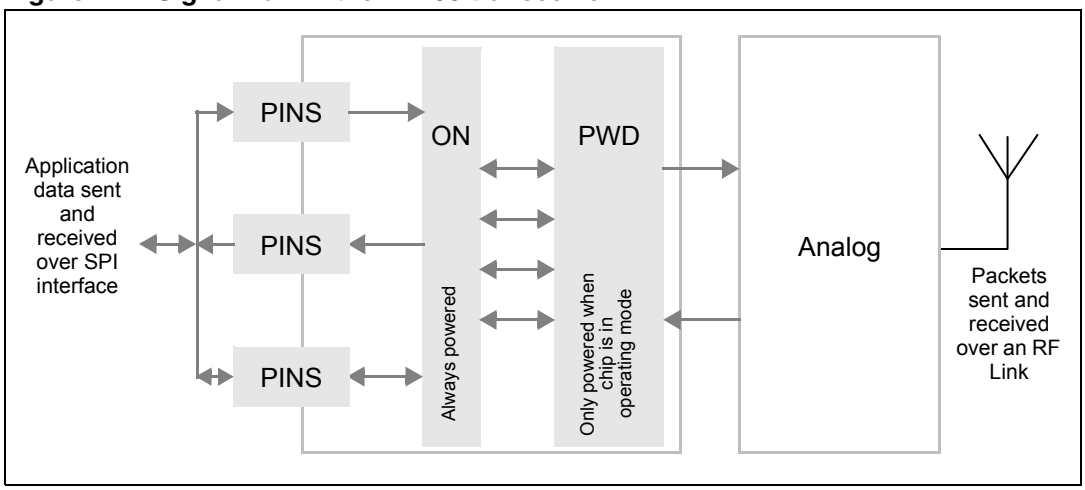
4.2.2 Digital part – PWD section

The PWD section, which is powered up only during operating mode, contains the remaining controls that are used when the chip is required for data transmission and reception.

4.3 Signal flow

Figure 2 illustrates the signal flow within the chip.

Figure 2. Signal flow in the TN100 transceiver



4.4 Commonly used set of register default values

The chip has been set with a number of default values that would be required for most standard applications to reduce the time required for chip initialization.

Note: A full listing of register default values is provided in Appendix B: Default register settings on page 224.

4.5 Programmable pull-resistors

The TN100 transceiver uses programmable pull-resistors to lower the cost of the bill of materials. The following pads can now be set as either pull-up or pull-down:

- SpiClk (pin 15)
- SpiSsn (pin 16)
- SpiTxd (pin 17)
- SpiRxd (pin 18)
- POnReset (pin 30)
- TxRx (pin 9)
- µCIRQ (pin 27)
- µCReset (pin 28)
- Digital IO pads (pins 19 to 22)

5 Pin description

This section provides a brief overview of the location and function of each pin.

Figure 3. Pinout

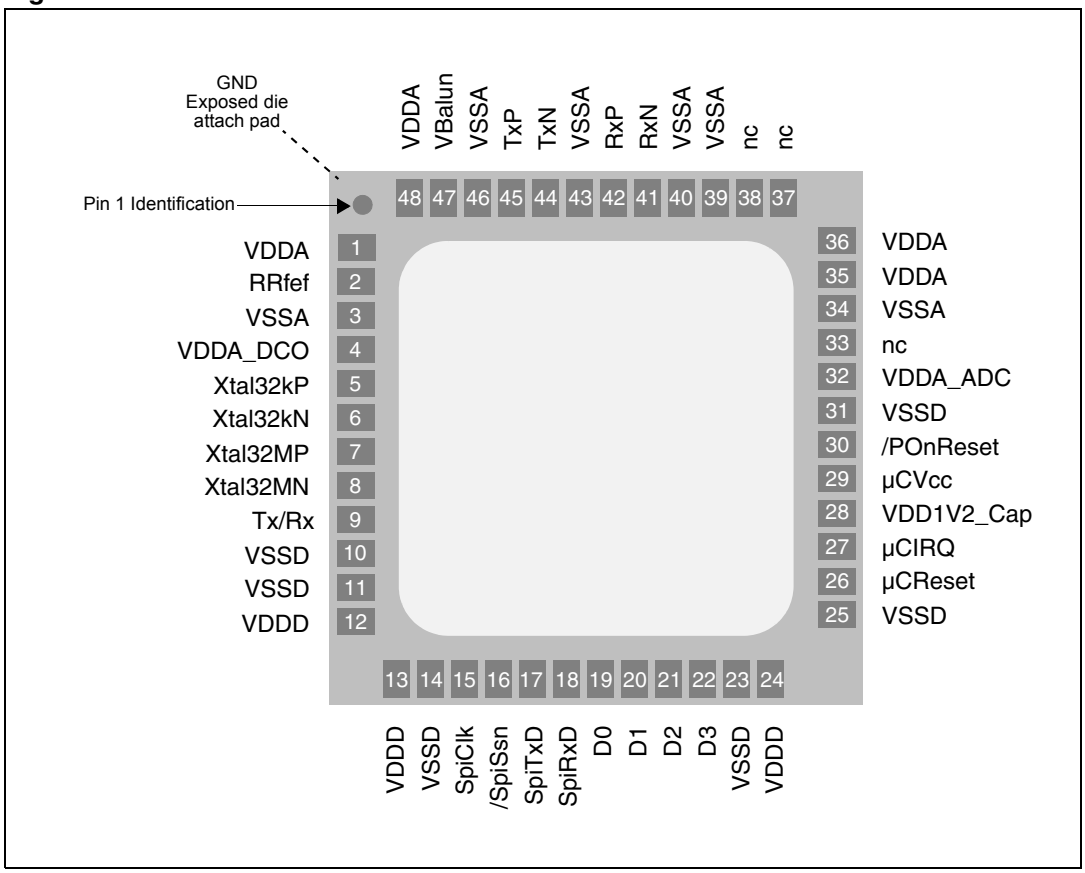


Table 1. Pin descriptions

Pin	Signal	Direction	Description
1	VDDA	Supply	Power supply for analog parts
2	RRef	Analog IO	Pin for external reference resistor
3	VSSA	Supply	Power supply for analog parts
4	VDDA_DCO	Supply	Power supply for DCO
5	Xtal32kP	Analog IO	Crystal 32768 Hz, input from external 32768 Hz frequency reference
6	Xtal32kN	Analog IO	Crystal 32768 Hz
7	Xtal32MP	Analog IO	32.0 MHz crystal oscillator, input from external 32 MHz frequency reference
8	Xtal32MN	Analog IO	32.0 MHz crystal oscillator
9	Tx/Rx	Digital output	Output signal to distinguish between transmit and receive
10	VSSD	Supply	Power supply for digital parts
11	VSSD	Supply	Power supply for digital parts

Table 1. Pin descriptions (continued)

Pin	Signal	Direction	Description
12	VDDD	Supply	Power supply for digital parts
13	VDDD	Supply	Power supply for digital parts
14	VSSD	Supply	Power supply for digital parts
15	SpiClk	Digital input	SPI clock
16	SpiSSn	Digital input	SPI slave selected
17	SpiTxD	Digital output	SPI transmit data (MISO)
18	SpiRxD	Digital Input	SPI receive data (MOSI)
19	D0	Digital IO	Digital input or output line 0
20	D1	Digital IO	Digital input or output line 1
21	D2	Digital IO	Digital input or output line 2
22	D3	Digital IO	Digital input or output line 3
23	VSSD	Supply	Power supply for digital parts
24	VDDD	Supply	Power supply for digital parts
25	Test	Digital IO	Pin for digital tests
26	µCReset	Digital output	Reset for external microprocessor
27	µCIRQ	Digital output	Interrupt request to external microprocessor
28	VDD1V2_CAP	Supply	1.2 V digital power supply decoupling.
29	µCVCC	Supply	Switchable power supply for external microcontroller
30	POnReset	Digital input	Power on reset line
31	VSSD	Supply	Power supply for digital parts
32	VDDA_ADC	Supply	Power supply for analog parts (Rx ADC)
33	TestCom	LF signal	Test pin for analogue signals
34	VSSA	Supply	Power supply for analog parts
35	VDDA	Supply	Power supply for analog parts
36	VDDA	Supply	Power supply for analog parts
37	TestRxP	LF Signal	Test pin for RX signals
38	TestRxN	LF Signal	Test pin for RX signals inverted
39	VSSA	Supply	Power supply for analog parts
40	VSSA	Supply	Power supply for analog parts
41	RxN	RF input	Differential receiver input (inverted)
42	RxP	RF input	Differential receiver input
43	VSSA	Supply	Power supply for analog parts
44	TxN	RF output	Differential transmitter output (inverted)
45	TxP	RF output	Differential transmitter output
46	VSSA	Supply	Power supply for analog parts
47	VBalun	Supply	Power supply for external balun (DC for TxP/TxN; needs to be blocked with an external capacitor in the 27pF to 47pF range close to the balun)
48	VDDA	Supply	Power supply for analog parts

5.1 Input and/or output pins

Table 2 describes the digital pins and analog pins of the TN100 transceiver.

Table 2. Digital input pins

Pin	Pin no.	Direction	Description
Xtal32kP	5	Input	Analog pin. 32.768 kHz crystal oscillator pin 1 or input for an external 32.768 kHz clock generator. Used to connect crystal or active frequency reference.
Xtal32kN	6	Output	Analog pin. 32.768 kHz crystal oscillator pin 2.
Xtal32MP	7	Input	Analog pin. 32 MHz crystal oscillator pin 1 or input for an external 32 MHz clock generator. Used to connect crystal or active frequency reference.
Xtal32MN	8	Output	Analog pin. 32 MHz kHz crystal oscillator pin 2.
/POnReset	30	Input	Power on reset signal.
SpiClk	15	Input	The SPI Clock is generated by the microcontroller (master) and synchronizes data movement in and out of the device through the pins SpiRxD and SpiTxD.
/SpiSSn	16	Input	SPI Slave Select (low active) is externally asserted before the microcontroller (master) can exchange data with the TN100 transceiver. Must be low before data transactions and must stay low for the duration of the transaction.
SpiRxD	18	Input	SPI Receive Data (MOSI).
SpiTxD	17	Output	SPI Transmit Data (MISO).
Tx/Rx	9	Output	Distinguishes between the TX and RX phase. Can also be used to provide an external power amplifier control. Active Low during TX, otherwise High.
µCReset	26	Output	Reset for external microprocessor.
µCIRQ	27	Output	Interrupt request to external microprocessor.
D0	19	Input/Output	Digital Input or Output (programmable, see configuration bits below), line 0.
D1	20	Input/Output	Digital Input or Output (programmable, see configuration bits below), line 1.
D2	21	Input/Output	Digital Input or Output (programmable, see configuration bits below), line 2
D3	22	Input/Output	Digital Input or Output (programmable, see configuration bits below), line 3. Note that a 32.768 kHz clock operates on this pin after reset/power up.
µCVcc	29	Output	Analog pin. Power supply for external microprocessor.

5.2 Configuring the digital I/O pins – D0 to D3 (pins 19 to 22)

Each digital I/O pin can be configured as either an input or an output pin. Signal levels or an alarm occurrence can be reported at a digital I/O pin that has been set as input. *Table 3* lists the fields are used for configuring digital I/O pins.

Table 3. Digital I/O pin configuration

Field	Offset	R/W	Description
DioDirection	0x04	WO	Controls the direction of Digital I/O port. Set it as either an input or an output pin.
DioOutValueAlarmEnable	0x04	WO	When a Digital I/O port is configured as input, this bit selects to be reported either the signal level at the port or the occurrence of an alarm.
DioAlarmStart	0x04	WO	Starts the alarm, and is set after the digital I/O port is configured to report the occurrence of an alarm.
DioAlarmPolarity	0x04	WO	When the digital I/O port is configured as an input that should report the occurrence of an alarm, then this bit is used to select the edge which should trigger the alarm. When the digital I/O port is configured as an output, then this bit selects whether the value programmed in DioOutValueAlarmEnable or the feature clock should be driven out of the digital I/O port.
DioUsePullup	0x04	WO	When set to true, a pull-up resistor is connected to the corresponding digital I/O pad.
DioUsePulldown	0x04	WO	When set to true, a pull-down resistor is connected to the corresponding digital I/O pad only, but when DioUsePullup is false.

Each digital I/O pin has one write strobe, as listed in *Table 4*.

Table 4. Digital I/O pin write strobe

Field	Offset	R/W	Description
DioPortWe	0x04	WO	Writes the settings of the 6 configuration bits to the digital I/O controller.

Each digital I/O pin has one status bit, as listed in *Table 5*.

Table 5. Digital I/O pin status

Field	Offset	R/W	Description
DioInValueAlarmStatus	0x04	RO	Each bit reports the signal level or the occurrence of an alarm at one of the four digital I/O ports, where bit 0 belongs to D0, bit 1 belongs to D1, bit 2 belongs to D2, and bit 3 belongs to D3.

5.3 Configuring the IRQ Pin – µCIRQ (pin 27)

The IRQ pin (µCIRQ) can be configured with either a high or low active polarity and can be driven as either push-pull or open-drain. The source of the interrupt can also be set.

The following fields are used to configure the IRQ pins as either low or high active, as well as either push-pull or open-drain:

Table 6. IRQ pin configuration

Field	Offset	R/W	Default
IrqPolarity (high/low active)	0x00	RW	Defines the polarity of the IRQ signal as either high or low active. The default is low active.
IrqDriver (push-pull/open-drain)	0x00	RW	Switches between push-pull or open-drain for IRQ output driver. The default is open-drain.

The following fields are used to drive the interrupt of the IRQ pin by either a transmitter interrupt, a receiver interrupt, a baseband timer interrupt, or a local oscillator interrupt:

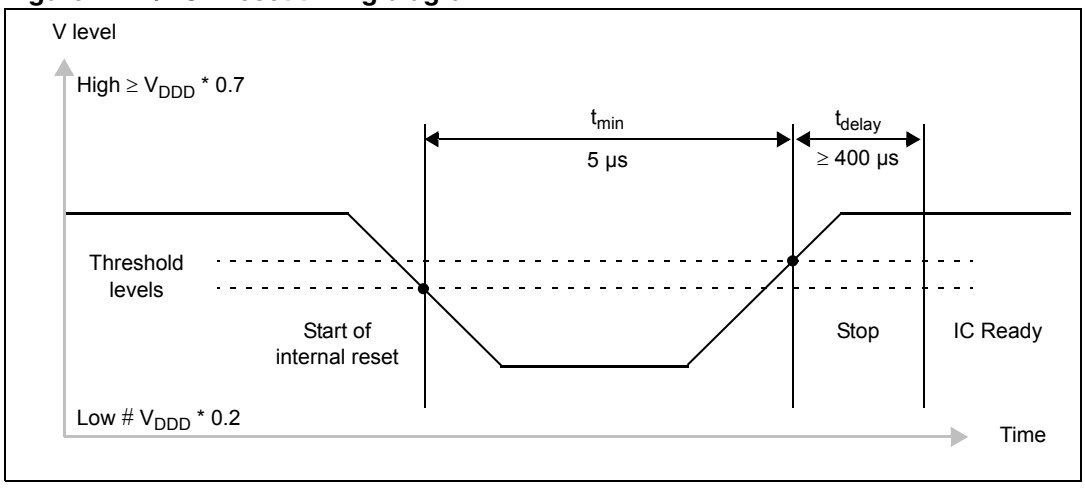
Table 7. Interrupts driving the IRQ pin

Field	Offset	R/W	Description
TxIrqEnable	0x0F	RW	The transmitter interrupt can be enabled to drive the interrupt line. Default is disabled.
RxIrqEnable	0x0F	RW	The receiver interrupt can be enabled to drive the interrupt line. Default is disabled.
BbTimerIrqEnable	0x0F	RW	The baseband timer can be enabled to drive the interrupt line. Default is disabled.
LoIrqEnable	0x0F	RW	The Local Oscillator interrupt can be enabled to drive the interrupt line. Default is disabled.

5.4 Power-on reset – /POnReset (pin 30)

/POnReset signal is active low. *Figure 4* shows a timing diagram for /POnReset.

Figure 4. /POnReset timing diagram



6 Memory map

This section describes the memory map of the TN100 transceiver. Procedures are provided for accessing the chip's programmable Register Block, as well the Baseband RAM, Chirp Sequencer RAM, and Correlator RAM. *Figure 5* shows the memory map of the TN100 transceiver.

Figure 5. Memory map

	Address	Memory Type 11-10	Page Select 9-8	Register Select 7	Bits 6-0
Correlator RAM	Hex FFF	11	11	1	Correlator RAM Page 3
				0	Register Block
			10	1	Correlator RAM Page 2
				0	Register Block
			01	1	Correlator RAM Page 1
				0	Register Block
			00	1	Correlator RAM Page 0
	Hex C00			0	Register Block
Chirp Sequencer	Hex BFF	10	11	1	Chirp Sequencer Page 3
				0	Register Block
			10	1	Chirp Sequencer Page 2
				0	Register Block
			01	1	Chirp Sequencer Page 1
				0	Register Block
			00	1	Chirp Sequencer Page 0
	Hex 800			0	Register Block
Unused	Hex 7FF	01			
	Hex 400				
Baseband RAM	Hex 3FF	00	11	1	Baseband Page 3
				0	Register Block
			10	1	Baseband Page 2
				0	Register Block
			01	1	Baseband Page 1
				0	Register Block
			00	1	Baseband Page 0
	Hex 000			0	Register Block

Table 8 lists the sections provided in the memory map.

Table 8. Memory map section

Address	Memory type	Description
0x000 to 0x3FF	Baseband RAM	512 bytes of Baseband RAM stores both data payload and MAC header values, depending on buffer configuration settings.
0x800 to 0xBFF	Chirp Sequencer RAM	Delivers two sequences of 6-bit values that synthesize the I and Q signals of a symbol.
0xC00 to 0xFFF	Correlator RAM	Stores the reference and threshold values for detection. It is are initialized with the FDMA, 4 µs default detector matrix. When other symbols are needed this register must be programmed with the appropriate values. The threshold must be programmed with the appropriate value in any case (even for the default matrix).
0x000 to 0x07F	Register Block	128-byte programmable chip register block provides chip configuration settings and is mapped to the entire memory of the TN100 transceiver.

Note: For a description of the Chirp sequencer, see Section 10: Chirp sequencer (CSQ) on page 53. The Correlator RAM is described in Section 6.3: Correlator RAM access on page 26. The Register Block is described in Section 6.2: 128-byte programmable register block on page 25.

6.1 Selecting a memory address

There are two ways to access the TN100 transceiver memory map: direct and indirect access. The Register block is accessed using a direct access model. The Baseband, Chirp Sequencer and Correlator RAM blocks are accessed using an indirect access model.

To access one of these RAM blocks using an indirect access model through register *0x0E – Baseband memory access*:

1. Select the memory type (Baseband, Chirp Sequencer or Correlator) using bits `DeviceSelect`.
2. Select the page pointer using bits `RamIndex`.
3. A Read or Write operation can now be performed.

For example to write a value to the Chirp Sequencer RAM at location 0x185, write in register *0x0E – Baseband memory access*:

1. Write `DeviceSelect = 0x2` to select the Chirp Sequencer
2. Write `RamIndex = 0x1` to select the RAM column 1.
3. Write to address 0x85.

For example to read a value from the Correlator RAM location at 0x280, write in register *0x0E – Baseband memory access*:

1. Write `DeviceSelect = 0x03` to select the Correlator RAM.
2. Write `RamIndex = 0x02` to select the Threshold.
3. Read from address 0x80.

Note: For a Read or Write operation at an address higher than 0x7F, the absolute address is always relative to the value stored in register 0x0E.

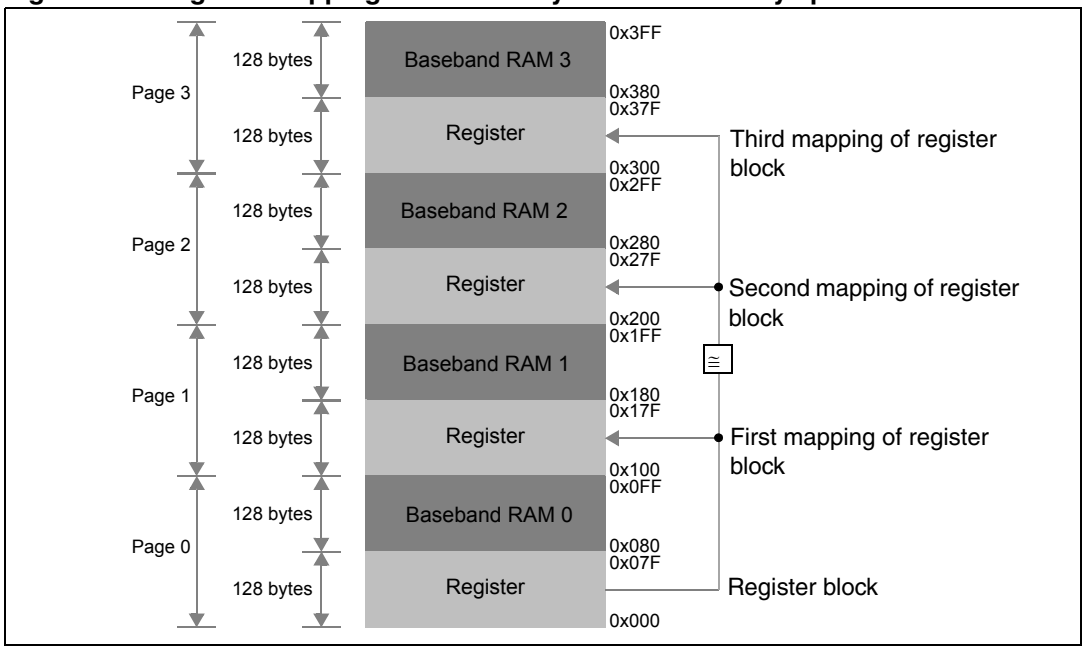
6.2 128-byte programmable register block

The TN100 transceiver provides a 128-byte programmable register block for chip configuration settings. The address space for the register is from 0x00 to 0xFF. However, it is mapped to three additional mapped registers within the 1024-byte memory space of the baseband RAM, where:

- Page 1 begins at offset 0x100
- Page 2 begins at offset 0x200
- Page 3 begins at offset 0x300.

These three mapped registers are logically equivalent to the register memory locations 0x00 to 0x7F. *Figure 6* illustrates this mapping.

Figure 6. Register mapping in the 1024-byte TN100 memory space



Note: All user accessible registers in this register block is fully described in Section 26.2: Description of chip registers on page 125.

6.2.1 Accessing a register address location

To access a memory location in the register only one SPI transfer is required, where:

- `SPI Address[7] = 0`
- `SPI Address[6:0] = <offset address in register>`

Either an SPI single byte operation or an SPI burst transfer can be used to access chip memory. The lower 7 bits of a given memory location are identical with the start address in the selected segment. SPI burst transfers are limited to 128 bytes (which is the segment size).

Note: A wraparound SPI burst transfer leads to unpredictable behavior. A wraparound SPI burst is when the number of bytes to be accessed is greater than the number of bytes from the start address to the end of the segment.

6.2.2 Setting a shadow variable for the RAM access register

When bit 7 of the 10 bit memory address is 1, the RamIndex field must be set with the two highest bits of this address (shifting right by 8 positions). The lower 8 bits are directly used in the next SPI access, which writes this data.

To reduce the overhead caused by writing the RamIndex field for each memory access, it is recommended that an RamIndex shadow variable be maintained in software. This variable can be used to back up the last value of the RamIndex field. If address locations in the same segment are accessed sequentially, write operations to the RamIndex field can be eliminated by comparing the RamIndex value with the shadow variable.

6.3 Correlator RAM access

The Correlator RAM contains the reference sequences of the detector and the detection thresholds. Use Correlator Memory I for programming the In-Phase values of the detector and use Correlator Memory Q for Quadrature-Phase values. Use Correlator Memory Thresholds to set the thresholds for In-Phase and Quadrature-Phase detection.

A Correlator RAM page is selected by setting DeviceSelect and RamIndex as shown in *Table 9*.

Table 9. Selecting a correlator RAM page

DeviceSelect setting	RAMIndex setting	Selected correlator memory page
DeviceSelect = 0x3	RamIndex = 0x0 (default value)	I (RamD3IPatI)
	RamIndex = 0x1	Q (RamD3IPatQ)
	RamIndex = 0x2	Thresholds (RamD3IThresholds)
	RamIndex = 0x3	Unused

6.4 Chirp sequencer RAM access

The Chirp Sequencer (CSQ) RAM space contains the values for I and Q, which are used to calculate Upchirps and Downchirps. The CSQ is set with a default matrix that has a symbol duration of 4 µs (4000 ns) and a 22 MHz bandwidth.

A Chirp Sequencer RAM page is selected by setting DeviceSelect and RamIndex as shown in *Table 10*.

Table 10. Selecting a correlator RAM page

DeviceSelect setting	RAMIndex setting	Selected correlator memory page
DeviceSelect = 0x2	RamIndex = 0x0 (default value)	Column 0
	RamIndex = 0x1	Column 1
	RamIndex = 0x2	Column 2
	RamIndex = 0x3	Unused

Note: For more details about the Chirp Sequencer, see Section 10: Chirp sequencer (CSQ) on page 53.

7 Clocking structure

This section describes the four clocks provided by the TN100 transceiver: 32.768 kHz clock, SPI clock, 32 MHz baseband clock, and the Chirp Sequencer (CSQ) clock.

7.1 Overview

The TN100 transceiver provides the following four clocks:

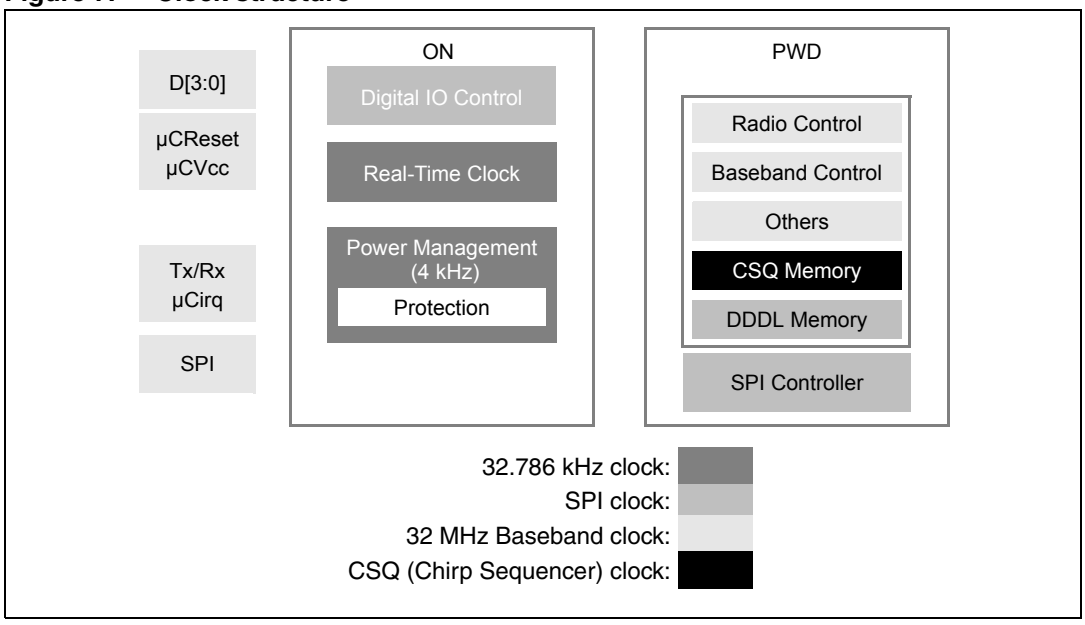
32.786 kHz clock – Used to run the real-time clock and power management.

SPI clock – Used for the SPI Controller and for the Digital IO Control used for running the four digital IO pins. The frequency of the SPI clock is dependent on the frequency required by the microcontroller. The maximum frequency is 27 MHz.

32 MHz baseband clock – Used for baseband control, radio control, and other baseband uses. The frequency of the baseband clock for the TN100 transceiver is 32 MHz and can be enabled or disabled by software.

CSQ (Chirp Sequencer) Clock – Used by the Chirp Sequencer (CSQ) Memory. The frequency of the CSQ clock is determined by dividing the Local Oscillator (LO) frequency by 10. The CSQ clock can be enabled or disabled by software.

Figure 7. Clock structure



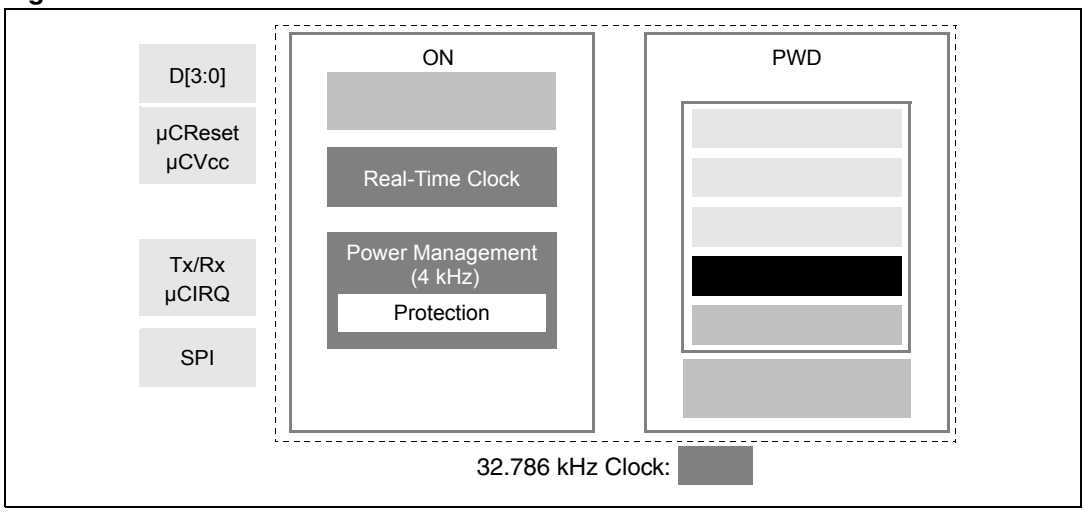
7.2 32.786 kHz real-time clock (RTC)

This real-time clock (RTC) runs at 32.768 kHz. It can be set or read through software using a 48-bit real-time clock value. As this clock is part of the ON section of the chip's digital part, it is always powered (unless the chip is completely powered off) and can, therefore, be used for creating a wake-up time event.

The real-time clock is also used to generate a 4 kHz clock (the Power Management clock) for use by the Power Management module in the ON section. Like the real-time clock, this slow clock is always available during power down mode to protect the connections between the SPI and the SPI Controller.

Figure 8 shows the real-time clock and Power Management modules that use the 32.768 kHz RTC.

Figure 8. 32.786 kHz real-time clock



7.2.1 Updating and reading the RTC through software

The 48-bit real-time clock is set and read through software. Since the real-time clock is updated every 1/(32.768 kHz), it is not directly accessible by the user. A RAM buffer `RamRtcReg` is used to hold the value of the internal real-time clock after it has been read. This buffer is also used to hold a value that will be written to the real-time clock.

Fields for updating RTC by software

Table 11 lists the fields used for updating the real-time clock with software.

Table 11. RTC and TimeB packets

Field	Offset	R/W	Description
`RtcCmdWr`	0x62	WO	Writes the 48-bit RTC value.
`RtcCmdRd`	0x62	WO	Reads the 48-bit RTC value
`RamRtcReg`	0xF0	RW	48-bit RTC value read from the RTC by software or written to the RTC by software

Updating the value of the RTC

To update the value of the real-time clock, do the following

1. Write a 48-bit value for the real-time clock to the buffer `RamRtcReg`.
2. Write this buffer to the real-time clock using the write command `RtcCmdWr`.

Reading the value of the RTC

To read the value of the real-time clock, do the following:

1. Read the real-time clock using the read command `RtcCmdRd` to place the 48-bit value of the real-time clock in the buffer `RamRtcReg`.
2. Read out the buffer `RamRtcReg`.

7.2.2 Manually or automatically updating the RTC using TimeB packets

The real-time clock can also be updated from TimeB packets that have been sent from a base station or other stations in a network. This updating can be performed automatically or manually.

If the updating is performed automatically, then when a TimeB packet is received, the RTC value in the packet is automatically written to the real-time clock.

If the updating is performed manually, then when a TimeB packet is received, the RTC value in the packet must be manually updated as described in *Section 7.2.1: Updating and reading the RTC through software on page 28*.

Fields for updating RTC by TimeB packets

Table 12 lists the fields used for updating the real-time clock by TimeB packets.

Table 12. RTC and TimeB packets

Field	Offset	R/W	Description
`RtcTimeBAutoMode`	0x62	WO	When set to 1, the RTC value is transferred in TimeB packets
`RtcTimeBRxAdj`	0x61	WO	Adjusts the RTC value for receiver delay.
`RtcTimeBTxAdj`	0x60	WO	Adjusts the RTC value for transmitter delay
`RamRtcTx`	0xE0	RW	The 48-bit RTC value that is to be transmitted (loaded/written) in a TimeB packet
`RamRtcRx`	0xE8	RW	The 48-bit RTC value that has been received in a TimeB packet

Manually updating the RTC

To manually update the real-time clock through the RTC value in a TimeB packet, enable Manual mode for TimeB packets by setting:

```
RtcTimeBAutoMode = NA_RtcTimeBAutoModeOff_BC_C (0x0)
```

This causes the 48-bit RTC value in the received TimeB packets to be stored in the real-time clock buffer RamRtcReg, where it can then be written to the real-time clock using the write command `RtcCmdWr`.

Automatically updating the RTC

To automatically update the real-time clock through from the RTC value in a TimeB packet, enable Auto mode for TimeB packets by setting:

```
RtcTimeBAutoMode = NA_RtcTimeBAutoModeOn_BC_C (0x1)
```

This causes the RTC values in received TimeB packets to be automatically stored in the real-time clock.

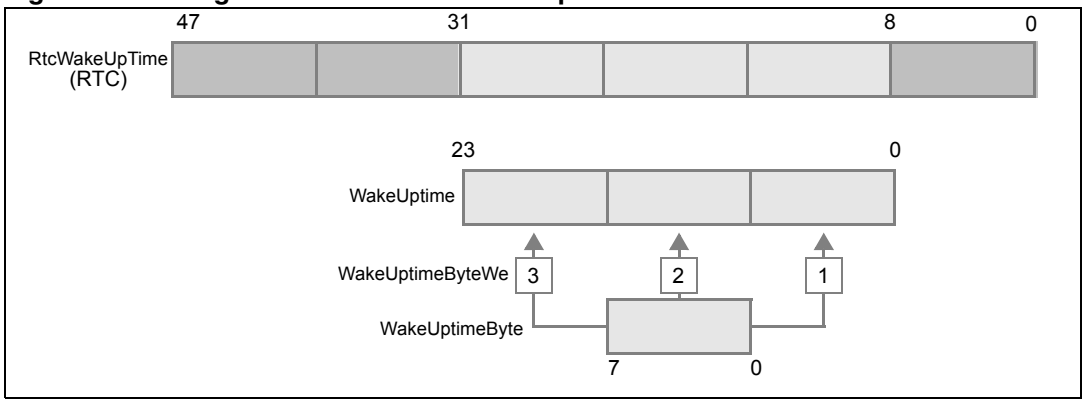
7.2.3 Using the RTC as wake-up event

The real-time clock can be used to create a wake-up time event at a predefined time. The field `EnableWakeUpRtc` enables the real-time clock to be used as a wake-up event.

The RTC wake-up time is set in `RtcWakeUpTime` but can only be accessed through the use of `WakeUpTimeByte` and `WakeUpTimeWe`.

The wake-up time is a 24-bit value split into 3 bytes. The field `WakeUpTimeByte` is used to set each of the three segments of the wake-up time (the segment to write is selected using the a byte-selector field `WakeUpTimeWe`). This wake-up time value is then compared to bits 31 to 8 of the real-time clock. When these values match, a wake-up event is then triggered. This process is shown in *Figure 9*.

Figure 9. Using real-time clock as wake-up event



Fields for setting RTC as wake-up event

Table 13 lists the fields using the real-time clock as a wake-up time event.

Table 13. Wake-up time fields

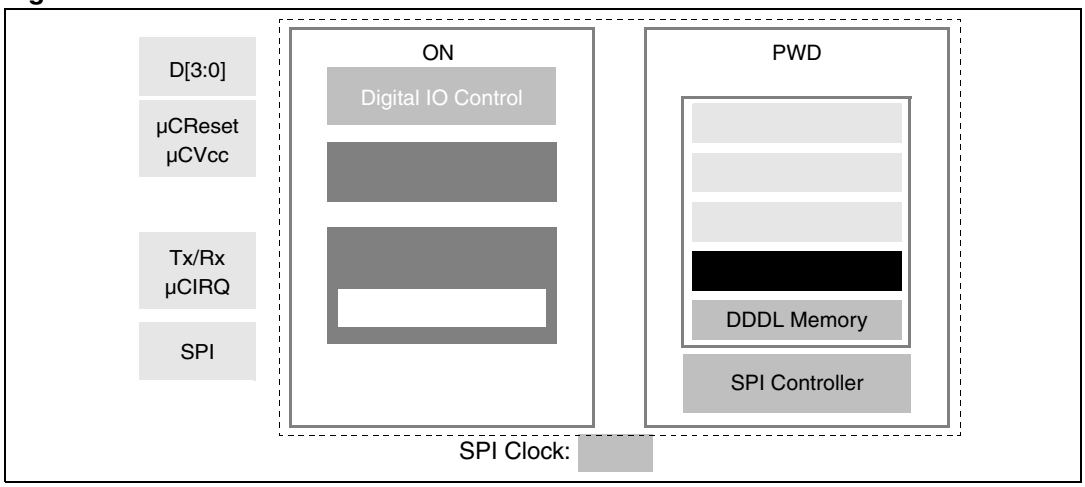
Field	Offset	R/W	Description
`WakeUpTimeByte`	0x01	WO	Stores a one byte value for the wake-up time, which is programed to the wake-up time RtcWakeUpTime using WakeUpTimeWe.
`WakeUpTimeWe`	0x02	WO	Loads the value of RtcWakeUpTimeByte to the appropriate byte of the wake-up time in the wake-up time circuitry.
`EnableWakeUpRtc`	0x06	RW	Enable real-time clock as Wake-Up Source

7.3 SPI clock

The SPI Clock is an externally delivered clock provided to the chip through an SPI signal from a master device, such as a microcontroller. The frequency of the SPI clock is dependent on the frequency of the master device. The maximum frequency of the SPI clock, however, is 27 MHz. It is provided as one of the four signals of the SPI interface: `SpiClk`.

Figure 10 shows the Digital IO Control and DDDL Memory modules as well as the SPI Controller that use the SPI clock.

Figure 10. SPI clock

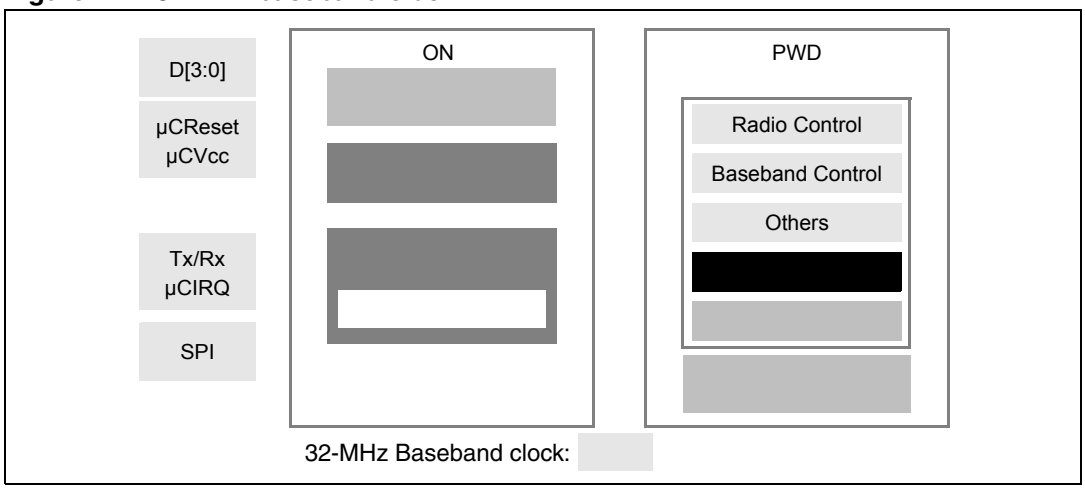


This clock is used to run the SPI Controller. As the Digital I/O Control is run by the SPI Controller, the SPI Clock is also used to run the four digital I/O pins: D0, D1, D2 and D3. It also runs all registers that are written by the SPI Controller.

7.4 32-MHz baseband clock

The baseband clock distribution obtains its frequency from the internal quartz oscillator. In the TN100 transceiver, the internal quartz oscillator provides a 32-MHz frequency. This clock is used for baseband control, radio control, and other baseband uses.

Figure 11. 32-MHz baseband clock



Note: The baseband clock can be enabled and disabled by software.

As the baseband clock is within the PWD section, this clock and all its modules run by this clock are powered off when the chip goes into PowerDownModeFull. Consequently, the contents of all chip registers are also lost.

Table 14. Baseband clock related fields

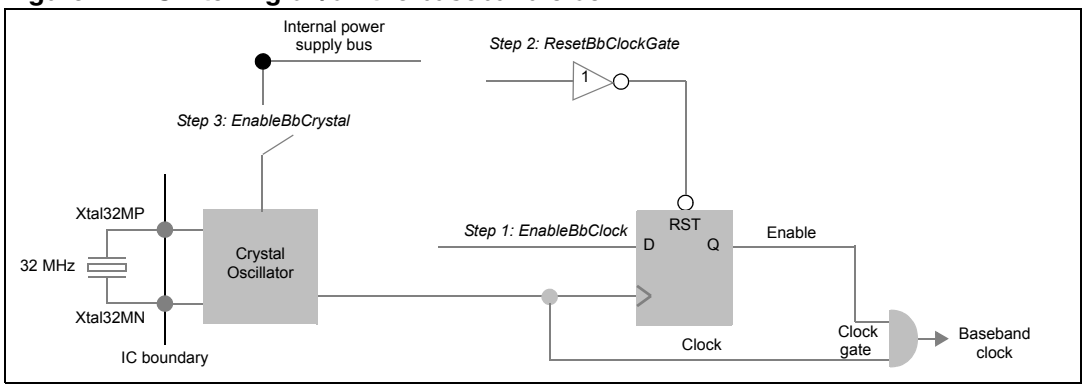
Field	Offset	R/W	Description
`ResetBbClockGate`	0x07	RW	Resets the baseband clock distribution circuitry.
`ResetBbRadioCtrl`	0x07	RW	Resets the digital baseband and radio control circuitries.
`EnableBbCrystal`	0x08	RW	Powers on the internal baseband oscillator.
`EnableBbClock`	0x08	RW	Enable the baseband clock distribution.

7.4.1 Stopping / enabling the 32-MHz baseband clock

To reduce power consumption without losing the contents of chip registers, the baseband clock distribution can be stopped and the quartz oscillator circuitry can be powered down.

Figure 12 shows the 32-MHz baseband clock and baseband quartz oscillator circuitry.

Figure 12. Switching on/off the baseband clock



Keeping clock switcher in stable state

The clock switcher is kept in a stable state during warm-up and shut down of the quartz oscillator by using `ResetBbClockGate`.

Stopping the 32-MHz baseband clock distribution

To stop the baseband clock distribution, do the following:

1. Set `EnableBbClock = 0` to switch off the baseband clock distribution.
2. Set `ResetBbClockGate = 1` to enable active reset of the baseband clock circuitry.
3. Set `EnableBbCrystal = 0` to power off the internal baseband quartz oscillator.

Enabling the 32-MHz baseband clock distribution

To enable the baseband clock distribution, do the following:

1. Set `EnableBbCrystal = 1` to power on the internal baseband quartz oscillator.
2. Set `ResetBbClockGate = 0` to inactivate the reset of the baseband clock circuitry.
3. Set `EnableBbClock = 1` to switch on the baseband clock distribution.

7.5 CSQ (Chirp Sequencer) clock

The Chirp Sequencer clock (CSQ) is required for programming the Chirp Sequencer RAM memory. The frequency of the CSQ clock is determined by dividing the Local Oscillator (LO) frequency by 10 (see *Section 21.2: Calibrating the local oscillator frequency on page 92*). The CSQ clock can be enabled or disabled by software using `EnableCsqClock`.

Figure 13 shows the CSQ Memory module that uses the Chirp Sequencer clock.

Figure 13. Chirp sequencer clock

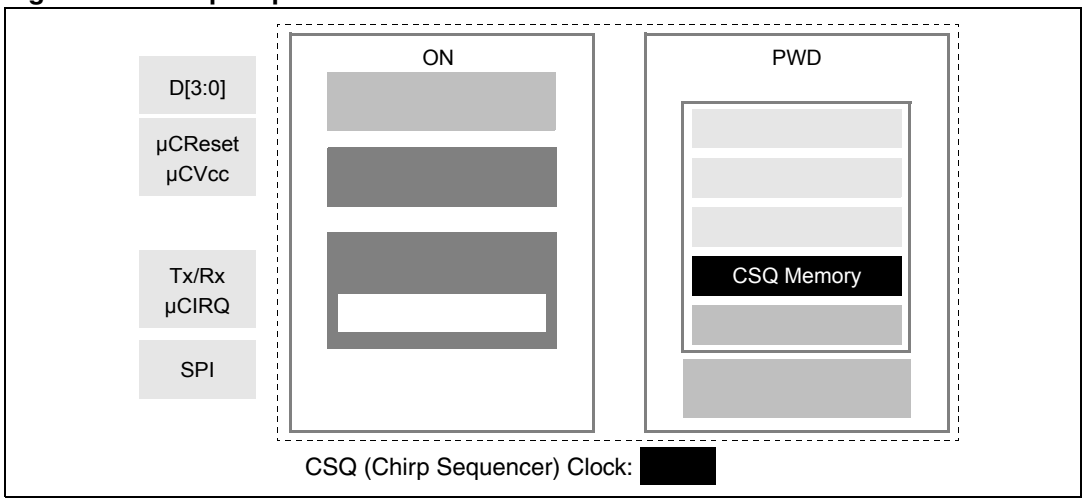


Table 15. Chirp sequencer clock related fields

Field	Offset	R/W	Description
`EnableCsqClock`	0x42	RW	Enables the Chirp Sequencer (CSQ) clock.

7.5.1 Stopping / enabling the Chirp Sequencer clock (CSQ)

Enabling the Chirp sequencer clock

1. LOenable=1
2. LODiv10enable=1
3. EnableCsqClock=1

Disabling the Chirp sequencer clock

1. EnableCsqClock=0
2. LODiv10enable=0
3. LOenable=1

7.5.2 Using the default matrix for transmission

To use the default matrix for transmission, set `CsqUseRam = 0`.

Note: '0' is the default value of `CsqUseRam`.

8 Power management

This section describes the five power management states of the TN100 transceiver, which are PwrDownModeFull, PwrDownModePad, PowerUp, Standby, and Ready.

The TN100 transceiver is designed to efficiently use the power required to operate the chip. Modules required during transmission and reception, such as the baseband memory, CSQ memory, the SPI Controller, and other modules are located in a section of the chip that can be powered down when not in use. To keep the chip operational, the minimum required modules are located in a section of the chip that uses minimal power to maintain a connection with the microcontroller.

8.1 Power management states

Table 16 shows the current consumption and activation time of the power management states.

Table 16. Power management states

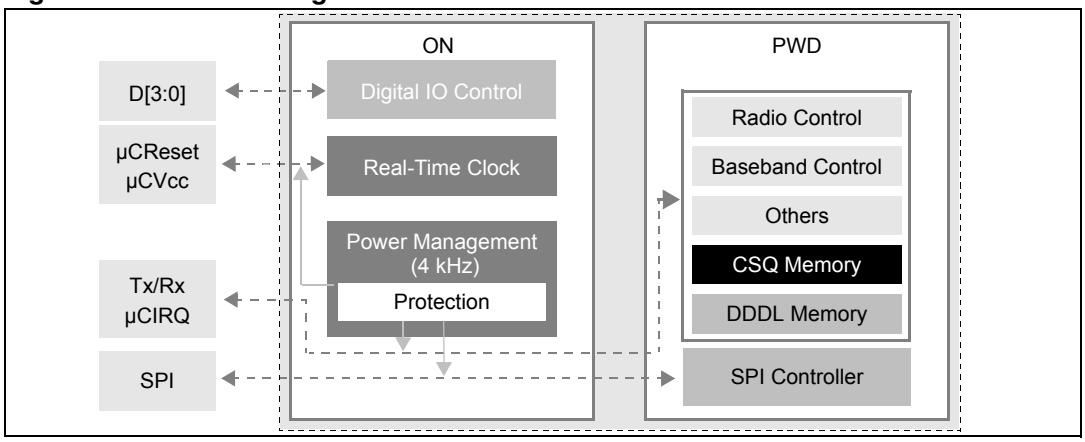
State	Current consumption(1)	Activation time into state
PwrDownModeFull	under 2.5 µA	PowerUp: between 1 and 32 ms (programmable) + boot time of the external microcontroller
PwrDownModePad	# 600 µA	PowerUp: between 1 and 32 ms (programmable) + boot time of the external microcontroller
PowerUp	# 700 µA	StandBy: # 5 ms (depending on the speed of the baseband quartz oscillator)
StandBy	# 2.5 mA	Ready (without reconfiguration) 6 µs(2)
Ready	# 4 mA	n./a.

1. Current consumption values are typical values only.
2. @ 4 Mbit/s SPI.

8.2 Power management module – ON and PWD sections

The Power Management module is located within a section called the “ON section.” It is provided with a clock frequency of 4 kHz by the 32.768 kHz RTC. *Figure 14* shows the ON and PWD sections of the TN100 transceiver, with the modules contained within each section. The connections between the pins and the ON and PWD sections are shown as dashed lines, while connections protected by the Power Management module protects are shown by the light lines.

Figure 14. Power management module



The Power Management module is responsible for maintaining the pull-up and pull-down states of the SPI interface during all power down modes. This includes the pins: `SpiClk`, `SpiSSn`, `SpiRxD`, and `SpiTxD`. This module is also responsible for maintaining the pull-up and pull-down states of several additional input/output pins, including:

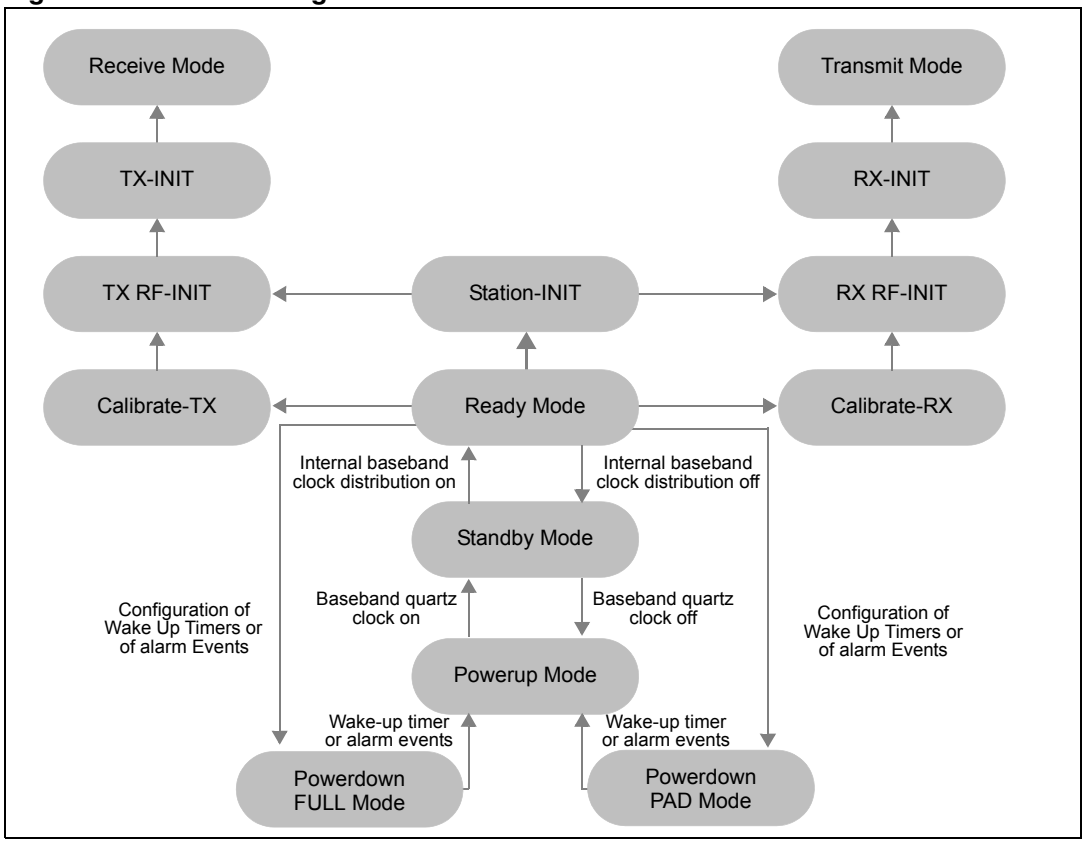
- `/POnReset`, an input pin which provides a signal to the power up reset line.
- `TxRx`, an output pin which is used to control an external power amplifier.
- `µCReset`, an output pin which resets an external microcontroller.
- `µCIRQ`, an output pin which sends interrupt requests to an external microcontroller.

The Digital IO pins are always powered as they can be programmed to provide a wake-up event to bring the PWD section back to powered up state.

8.3 Power management state model

The power management states model of the TN100 transceiver is shown in *Figure 15*.

Figure 15. Power management state model



8.4 Power management fields

Table 17 lists the fields used for Power Management.

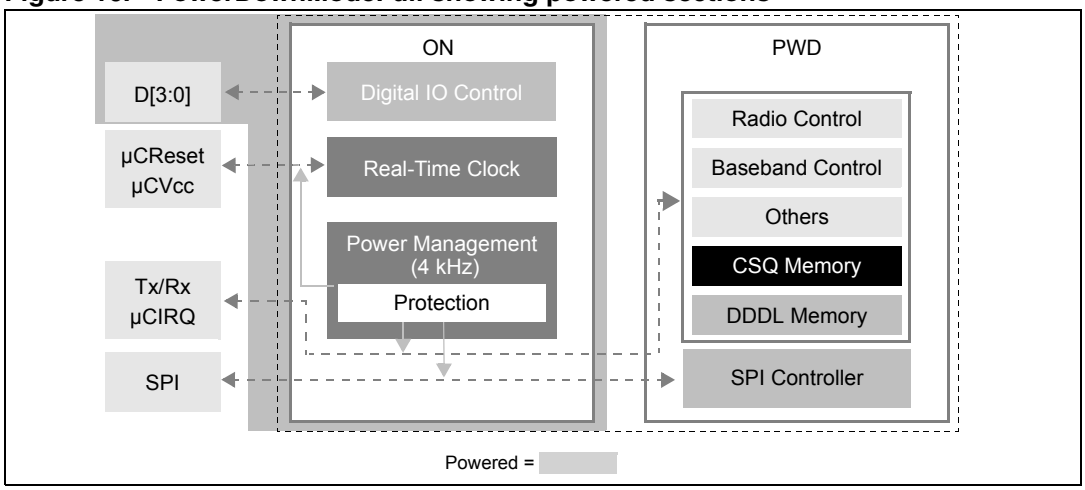
Table 17. Power management related fields

Field	Offset	Description
`EnableWakeUpRtc`	0x06	RTC wake-up event brings the chip into the power up state.
`EnableWakeUpDio`	0x06	Alarm event (rising/falling edge) at a digital IO configured as a wake-up source (see register 0x4) brings the chip into the power up state.
`PowerUpTime`	0x06	Configures duration of the power-up time (duration of active µCReset).
`PowerDownMode`	0x06	Selects the power down mode to use when the chip enters the powered down state
`PowerDown`	0x07	Brings the chip to the configured power-down state. This bit will be automatically cleared when the chip is powered-up.

8.5 PowerDownModeFull state

When the chip is in PowerDownModeFull, it consumes less power but after power up the chip will need to be reconfigured. The pins and registers of the chip are powered off but reconfiguration of the registers is required after wake-up. Also leakage current is the lowest possible in this mode.

Figure 16. PowerDownModeFull showing powered sections



Current Consumption

The current consumption[(a)] during PowerDownModeFull is under 2.5 µA.

Chip State

The chip is in the following state during PowerDownModeFull:

Table 18. PowerDownModeFull state

Module	State
External Microcontroller (µcVcc)	Powered down
PWD section	Powered down (all register settings lost)
ON section	Powered
Real-time clock	Running
Wake-up timer	Running
Digital IO Control	Running
Power Management module	Running

Setting PowerDownModeFull

To set `PowerDownModeFull`, do the following:

1. Set `PowerDownMode = NA_PowerDownModeFull_C (0x0)`
2. Set `PowerDown = 1`

a. Current consumption values are typical values only and are values without an external microcontroller and where digital IO pads are tri-state or inputs and no external pull-up resistors soldered on pads.

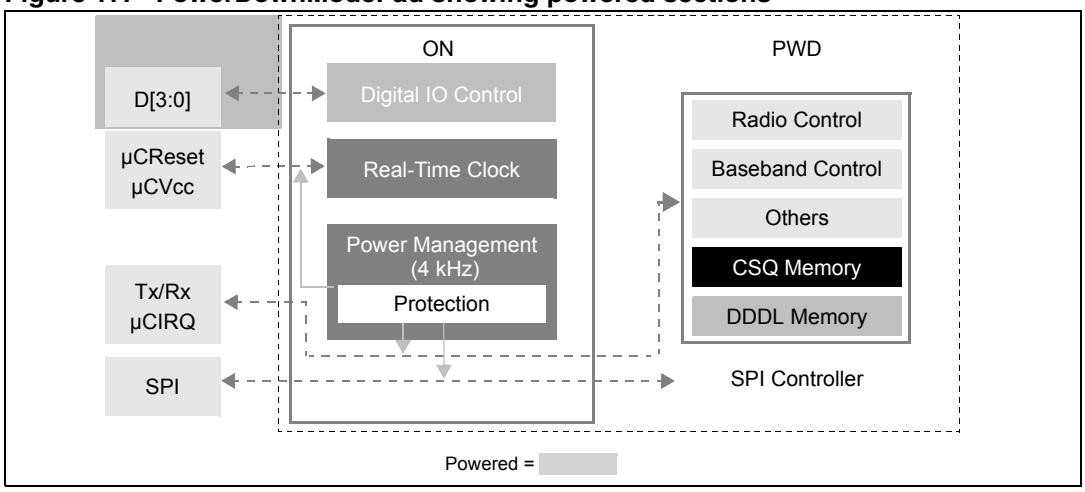
Bringing to PowerUp State

Either an internal wake-up timer or an external event on a Digital IO pin brings the chip out of PowerDownModeFull into PowerUp state. This activation time is programmable using the field PowerUpTime in a range between 1 and 32 ms + boot time of the external microcontroller. After the chip is in PowerUp mode, the external microcontroller can then initialize and (optionally) calibrate the chip.

8.6 PowerDownModePad state

When the chip is in PowerDownModePad, it consumes more power than PowerDownModeFull but after it has been powered up, it does not need to be reconfigured. In this state, all output pads are disabled and all bi-directional pads are switched to input, but all transceiver registers are still powered. Reconfiguration of the chip registers is not required, but leakage current is higher.

Figure 17. PowerDownModePad showing powered sections



Current Consumption

The current consumption during PowerDownModePad state is approximately 600 µA.

Chip State

The chip is in the following state during PowerDownModePad:

Table 19. PowerDownModePad state

Module	State
External Microcontroller (µcVcc)	Powered down
PWD section	Powered (all register settings maintained)
ON section	Powered
Real-time clock	Running
Wake-up timer	Running
Digital IO Control	Running
Power Management module	Running

Setting PowerDownModePad

To set PowerDownModePad, do the following:

1. Set `PowerDownModePAD = NA_PowerDownModePad_C (0x1)`
2. Set `PowerDown = 1`

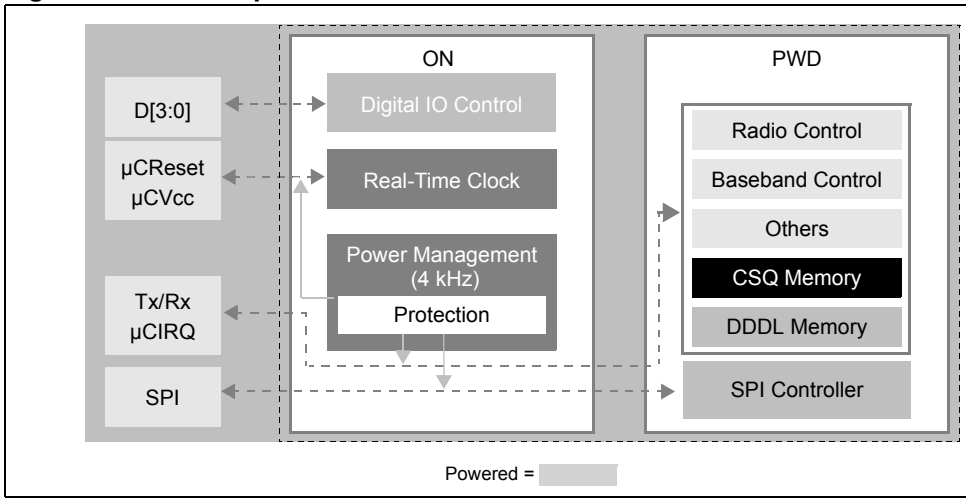
Bringing to PowerUp State

Either an internal wake-up timer or an external event on a Digital IO pin brings the chip out of PowerDownModePad. The chip then goes into PowerUp state. The activation time is programmable using the field `PowerUpTime` in a range between 1 and 32 ms + boot time of the external microcontroller. After the chip is set in Ready state, the external microcontroller can then initialize and calibrate the chip.

8.7 PowerUp state

The chip enters PowerUp state from either PowerDownModeFull or PowerDownModePad through on of the following: internal wake-up timer, external event on a Digital IO pin, or an external reset

Figure 18. PowerUp state



Current Consumption

The current consumption during Powerup state is approximately 700 µA.

Chip State

When the chip is brought to PowerUp state, the following occurs:

Table 20. PowerUp state

Module	State
External Microcontroller (µcVcc)	Powered
From PowerDownModeFull to PowerUp	PWD section powered and all pins are enabled
From PowerDownModePad to PowerUp	All output pins are enabled and digital IO pins are switched to bi-directional

Table 20. PowerUp state (continued)

Module	State
Real-time clock, Wake-up timer, and Power Management module	Running
Internal baseband quartz oscillator	Powered on and then baseband clock distribution is switched off
Internal baseband modules	Initiailzed with default settings and can now be programmed via the SPI interface

Bringing to Standby state

When the chip is brought from PowerUp state into Standby state, all blocks are powered up except the baseband distribution, which remains switched off. The activation time to bring the chip into Standby state from PowerUp state is approximately 5 ms, depending on the speed of the baseband quartz oscillator.

Wake-Up Sources for PowerUp state

The chip is brought to PowerUp state by a wake-up source. This is either a RTC wake-up event or an alarm event (a rising or falling edge) at one of the digital IOs configured as a wake-up source. *Table 21* lists the fields used to set a wake-up event.

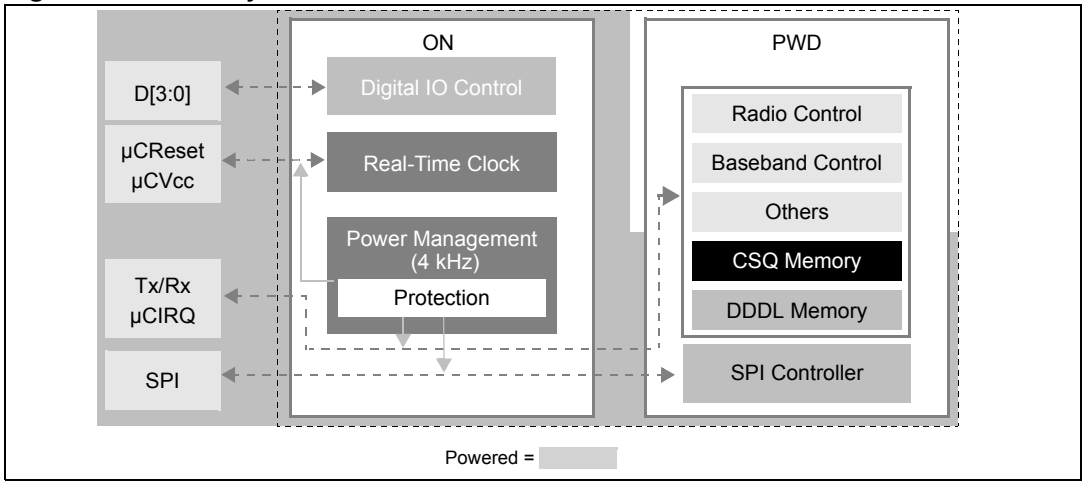
Table 21. Fields for setting a wake-up event

Field	Offset	Description
`EnableWakeUpRtc`	0x06	Enable the Real-time clock as a wake-up source. Default is disabled.
`EnableWakeUpDio`	0x06	Enable a digital IO pin as a wake-up source. Default is disabled.

8.8 Standby state

When the chip is In Standby state, only the baseband clock distribution is switched off. All other blocks of the chip are fully powered.

Figure 19. Standby state



Current Consumption

The current consumption during `Standby` state is approximately 2.5 mA.

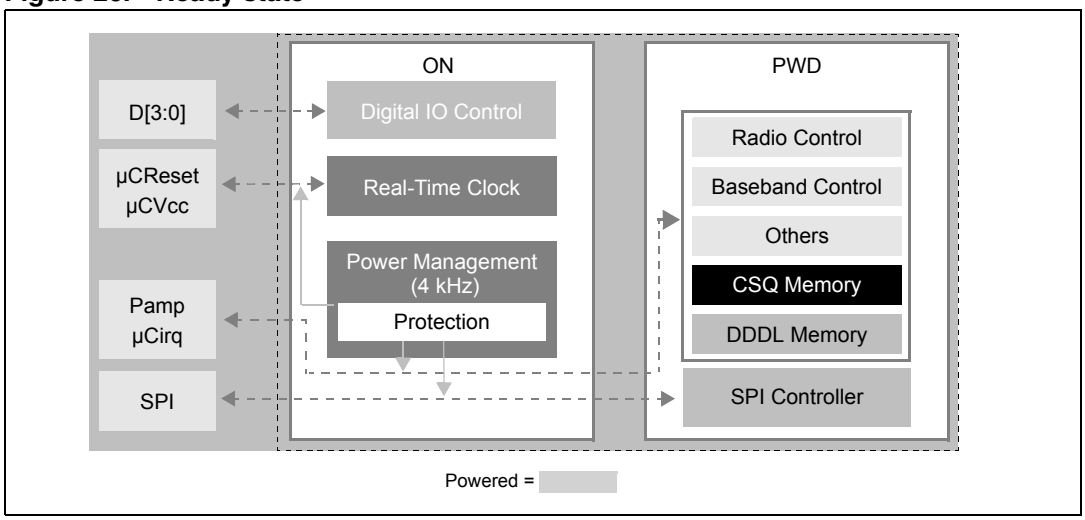
Bringing to Ready state

To bring the chip into Ready state, the baseband clock distribution is switched on. Activation time to bring the chip into Standby mode from PowerUp mode is 6 µs @ 4 Mbit/s SPI (without reconfiguration).

8.9 Ready state

When the chip is In Ready state, the baseband clock distribution is switched on and the microcontroller can immediately initiate a packet transmission or reception.

Figure 20. Ready state



Current Consumption

The current consumption during Ready mode is approximately 4 mA.

8.10 Powering off the chip

When the chip is completely powered off, which is when the power source is disconnected from the chip, both the ON and PWD sections are powered off and all chip settings are lost. The connection between the microcontroller (master device) and the chip (slave device) are lost and the Local Oscillator will require recalibration.

9 Programming interface (SPI)

This section describes the programming interface (SPI) to the TN100 transceiver, including descriptions of the SPI Controller, signals, interface connections, timing, and address format.

The TN100 transceiver is programmed through the use of a synchronous *Serial Peripheral Interface* (SPI) interface. This SPI interface is used to write control bits into memory locations in the chip register and baseband RAM locations. The SPI communicates via the SPI bus between a master device, such as a microcontroller, and one or more slave devices, including the TN100 transceiver and any serial memory devices.

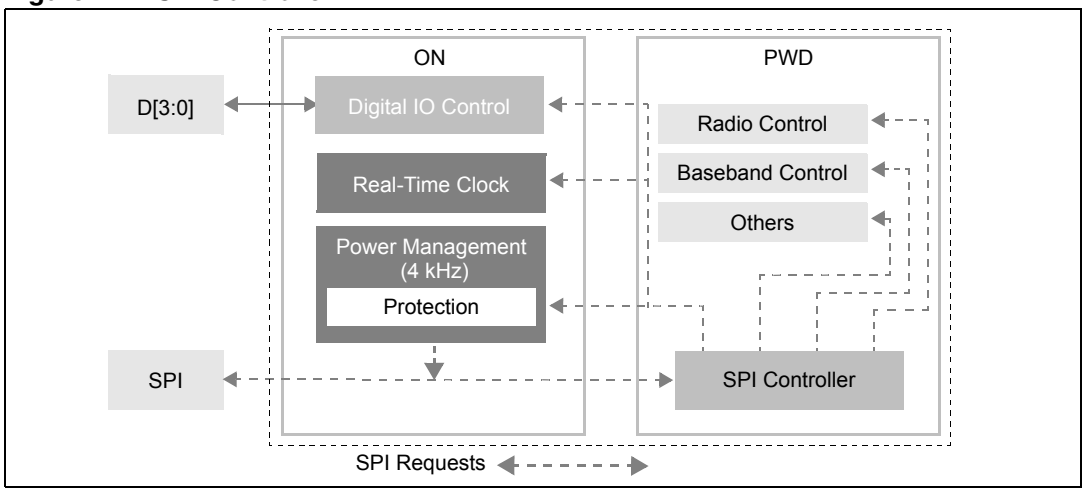
9.1 SPI controller

All SPI requests are distributed through the SPI Controller, whether in the PWD section or the ON section of the TN100 transceiver. In particular, the SPI Controller, which uses the SPI clock, directs SPI requests to the appropriate registers. All registers written by the SPI Controller are run with SPI clock, with the exception of Read Only registers.

Note: SPI requests are only possible when the PWD section of the chip is powered, as the SPI Controller is located in this section. See Section 9.1.1: SPI controller and power management on page 42.

Figure 21 shows the flow of SPI requests in the PWD and On sections.

Figure 21. SPI Controller



9.1.1 SPI controller and power management

PowerdownModeFull: When the chip is in PowerdownModeFull state the PWD section is completely powered down to reduce chip power consumption, which means all register settings are lost. However, the ON section retains a small amount of power to save specific chip settings related to the Digital IO pins, the real-time clock, and Power Management. Also, the connections between a microcontroller and the SPI Controller is protected to enable the chip to quickly wake-up and to be fully operational in a very short time.

PowerDownModePad: When the chip is in PowerDownModePad state, the PWD section remains powered, which means all register settings are maintained.

PowerUp: When the chip is in PowerUp state, the connection between the microcontroller and the SPI Controller over the SPI bus is quickly reestablished. Then, depending on what power down state was set, the microcontroller then initializes the registers in the PWD section and (optionally) calibrates the Local Oscillator frequency.

9.1.2 SPI controller and registers

The SPI Controller contains a mirror of the registers in the ON part. This is done to save power as less connections are maintained between the ON and PWD parts, especially for those registers relating to Digital IO control.

The SPI Controller also include those registers that must be accessible when the baseband clock is off, including:

- Pad controls – For example, registers for the SPI direction (MSB/LSB).
- Clock controls – For example, registers for enabling for baseband clock.
- Constants – For example, registers for the chip version and revision number.

9.2 SPI bit ordering

The SPI bit order of the TN100 transceiver is configurable, either LSB (Least Significant Bit First) or MSB (Most Significant Bit First). To set the bit order for the SPI bus, use the field `SpiBitOrder`, where:

- `SpiBitOrder = 1`: the first bit transmitted over the SPI is MSB.
- `SpiBitOrder = 0`: the first bit transmitted over the SPI is LSB.

The default bit order after reset or power-up is LSB first. *Table 22* lists the field used to set the SPI bit order.

Table 22. Bit order fields

Field	Offset	Description
`SpiBitOrder` (MSB/LSB first)	0x00	Sets the bit order for the SPI bus. Default is LSB.

Note: The byte order always remains the same.

9.3 SPI signals: SpiClk, SpiSsn, SpiTxD, SpiRxD

The SPI interface works as a half-duplex channel and uses the following four signals:

- SPI Clock – `SpiClk`

  This is a clock signal applied by the microcontroller (master device).
- SPI Slave Select – `/SpiSsn`

  This signal is used by the microcontroller (master device) to select the TN100 transceiver (slave device) for communication.

Note: This signal is specific to only one slave.

- Data Transmit – `SpiTxD`

  This signal is MISO (Master – Input; Slave – Output). With this signal, data is sent from a slave device (such as the TN100 transceiver or other slave devices) to a master device (microcontroller).

  `SpiTxD` is programmable as either push-pull or open-drain pin. When more than one slave device is connected to the SPI bus, the SpiTxD pin should be set to open drain mode. See also *Section 9.4: SPI TxD output driver on page 44*.

Note: In open drain mode an external pull up resistor is required.

- Data Receive – `SpiRxD`

  This signal is MOSI (Master – Output; Slave – Input). With this signal, data is sent from a master device (microcontroller) to a slave device (such as the TN100 transceiver or other slave devices).

9.4 SPI TxD output driver

The SPI TxD Output Driver can be set as either push-pull or open-drain by using `SpiTxDriver`.

Open-Drain: In this setting, the pad is driven only when a logic 0 is sent from the TN100 transceiver to the SPI master. Otherwise, the output is in high-impedance state. Do the following to set the SPI TxD output driver to open-drain:

- SpiTxDriver = 0

Push-Pull: The pad is driven in push-pull mode, but only when data is sent from the TN100 transceiver to the SPI master. Otherwise, the output is in high-impedance state. Do the following to set the SPI TxD output driver to push-pull:

- SpiTxDriver = 1

Default value: The default value is `SpiTxDriver = 0` (false).

Table 23 lists the field used to switch the TxD output driver between push-pull and open-drain.

Table 23. SPI TxD output driver field

Field	Offset	Description
SpiTxDriver	0x00	Switches between push-pull and open-drain for the TxD output driver (SpiTxD - pin 17).

9.5 SPI bus timing values

Table 24 lists the timing values for the SPI bus.

Table 24. SPI bus timing values

Parameter	Minimum	Maximum	Description
f_{max}	–	27 MHz	SpiClk
t_{LC}	18.5 ns	–	Low time SpiClk
t_{HC}	18.5 ns	–	High time SpiClk
t_{SS}	4 ns	–	/SpiSsn Setup
t_{HS}	2 ns	-	/SpiSsn Hold
t_{SRxD}	4 ns	–	SpiRxD Setup
t_{HRxD}	2 ns	–	SpiRxD Hold
t_{PDTxD}	–	18.5 ns	SpiTxD Propagation Delay Drive
t_{HTxD}	2.5 ns	–	SpiTxD Hold
t_{PTxDZ}	–	18.5 ns	SpiTxD Propagation Delay High Impedance

9.6 SPI transfer rate

9.6.1 Maximum transfer rate

The maximum transfer rate for communication between the microcontroller (the master device) and the TN100 transceiver (the slave device) is 27 MBit/s.

9.6.2 Bytes per transfer

For each SPI transfer, between 3 and 130 bytes (1-128 data bytes) can be sent per transfer.

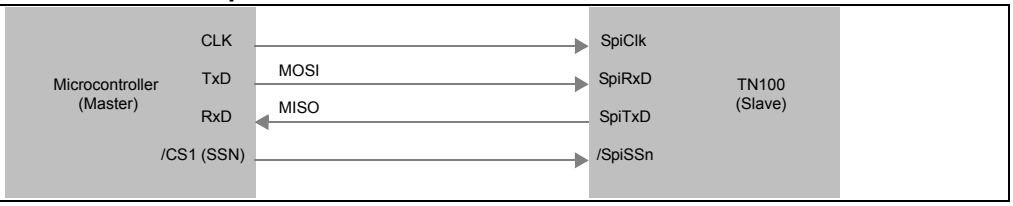
9.7 SPI interface connections

The TN100 transceiver can be wired with an SPI master device with a four-wire, a three-wire, or multiple-slave configuration.

9.7.1 Four-wire configuration

Four-wire configuration is the standard configuration between the TN100 transceiver (slave) and a microcontroller (master). It requires all four lines of the SPI interface, as illustrated in *Figure 22*.

Figure 22. Four-wire option of the SPI



9.7.2 Three-wire configuration

If only a single TN100 transceiver (slave) is connected to the microcontroller (master), then the SpiSSn input pin can be connected to GND.

This configuration saves one pin on the SPI master device or makes the pin available for other functions. This saves space on a circuit board, which is significant since many applications using the TN100 transceiver require a small form factor. Also, if the SpiSSn pin is driven with an active-low voltage level permanently, the power consumption of the TN100 transceiver is not increased.

9.7.3 Multiple slave configuration

If multiple slave devices (for example, multiple TN100 transceivers) are controlled by a single master device, the /`SpiSSn` pin of each slave device is connected to the `SSN` pin on the master device.

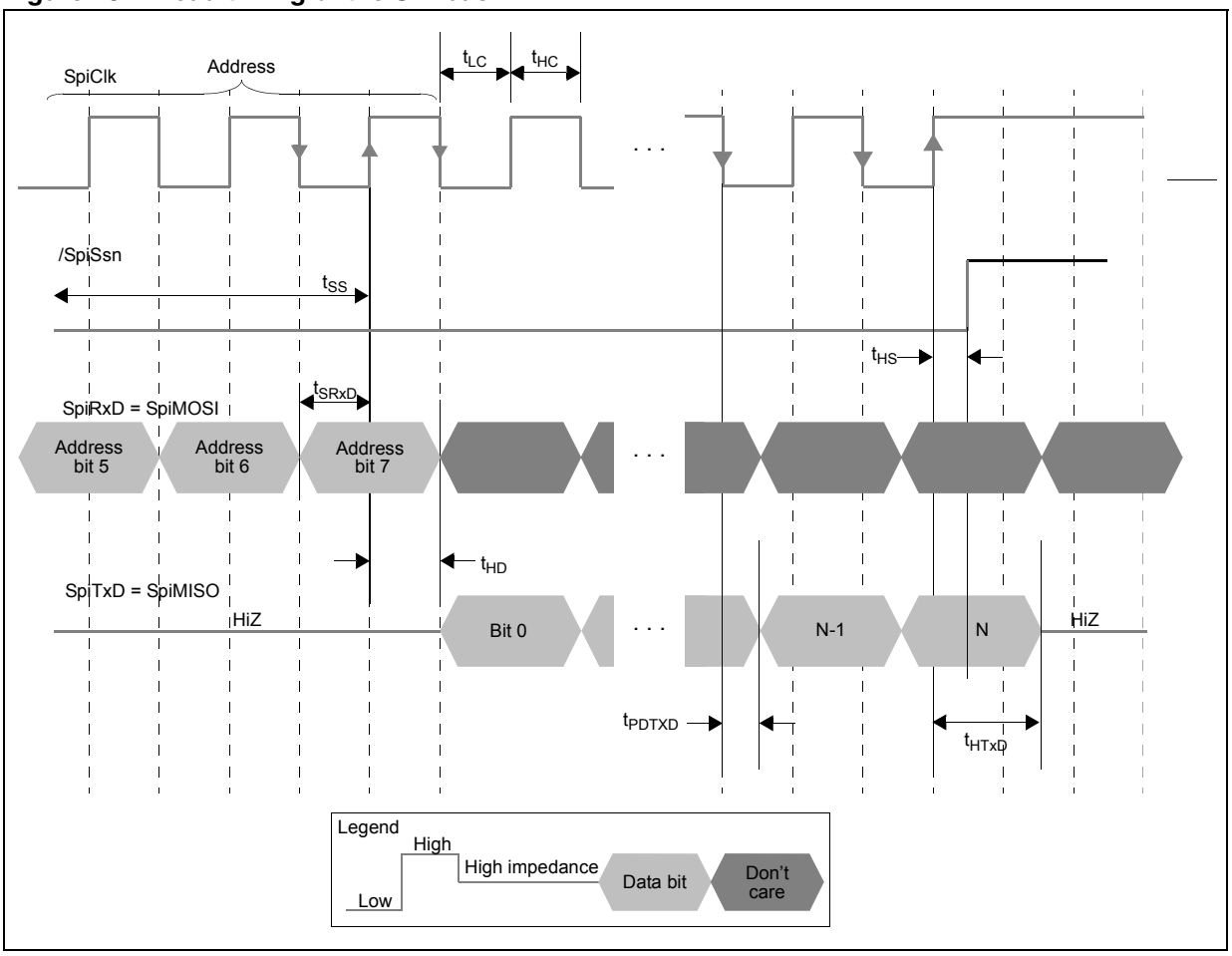
9.8 Read timing of the SPI bus

The read timing of the SPI bus is as follows:

1. After receiving the last address bit, the TN100 transceiver begins to switch the data bits to `SpiTxD`.
2. On each falling edge of `SpiClk`, a new data bit is assigned.
3. The output starts driving with the first read data bit.
4. The timing of the `SpiRxD` pin that is used for length and address during read is identical to the timing of the `SpiRxD` pin that is used during writing.

Figure 23 shows the read timing of the SPI bus.

Figure 23. Read timing of the SPI bus



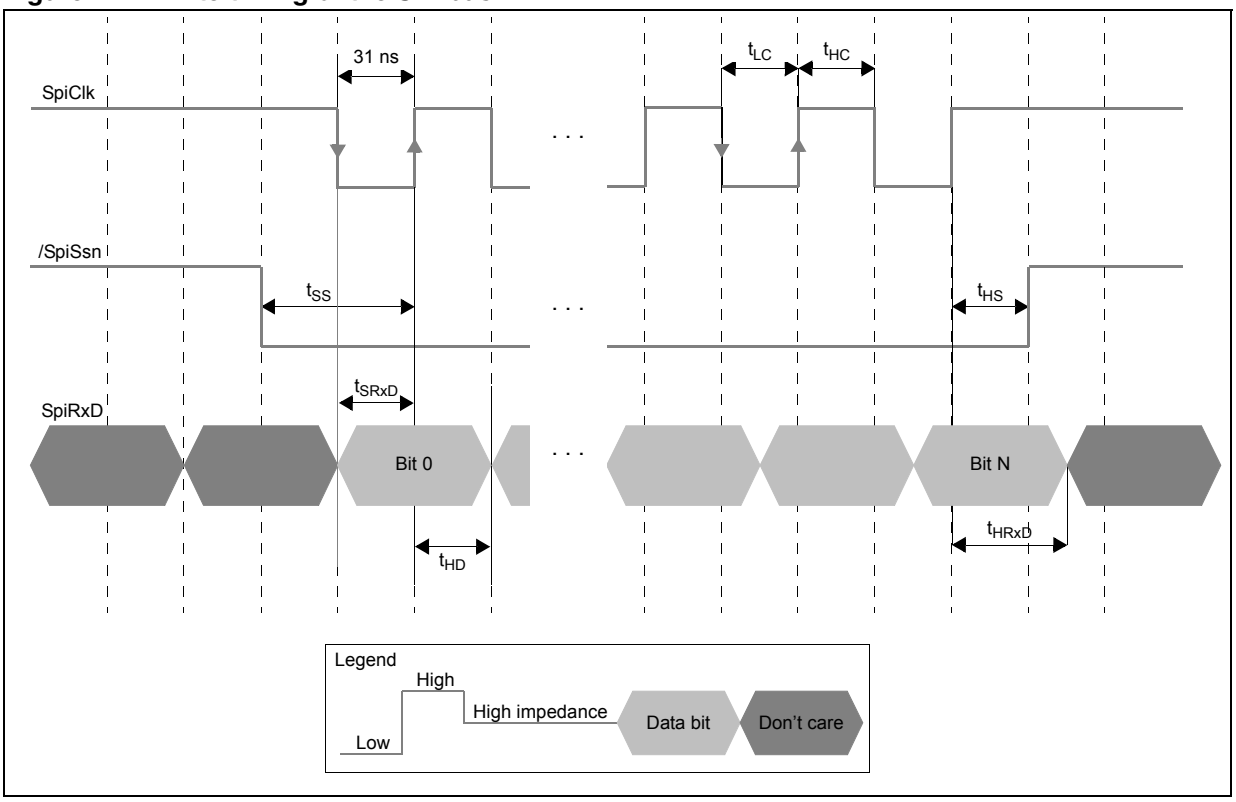
9.9 Write timing of the SPI bus

The write timing of the SPI bus is as follows:

1. On the first falling edge of the clock signal (`SpiClk`), with the master device SPI slave select (/`SpiSsn`) being low, the first data bit is switched (sent) to `SpiRxD`.
2. As long as /`SpiSsn` is activated, the TN100 transceiver latches one data bit onto each rising edge of the clock signal (`SpiClk`).
3. If several slave devices share the SPI bus, then the master device must control the communication with slave-dedicated chip selects (/`SpiSsn(n)`).
4. To complete an operation to a specific SPI slave, the /`SpiSsn` signal must be set to high after the last data bit has been transmitted.

Figure 24 shows the write timing of the SPI bus.

Figure 24. Write timing of the SPI bus



9.10 SPI address format

The SPI of the TN100 transceiver implements a burst capable address mode with automatic address increment. The SPI protocol has a variable data payload. *Figure 25* shows the SPI transfer format.

Figure 25. SPI transfer format

Byte 1: <Instruction>	RW	Length
Byte 2: <Address>	Start Address	
Byte N: 3 ≤ <N byte data> ≤ 130	Data	

9.10.1 Byte 1: instruction

The <Instruction> field has a fixed length of one byte and contains the command type (read[0] / write[1]) as well as the number of data bytes:

MSB LSB

7	6	5	4	3	2	1	0
Direction Bit	Data Bytes Size						

- Bit 7 is the “direction bit” and is used to differentiate between read or write access, where:
  - Write access = 1
  - Read access = 0
- Bits 0 to 6 provides the data byte size. This is the size of data that will be sequentially written to the TN100 transceiver or read from the TN100 transceiver in an access operation. The maximum length is 128 bytes, which is indicated by a 0x00 value.

9.10.2 Byte 2: address

The <Address> field has a fixed length of one byte and contains the first address to be accessed. Additional data bytes are accessed at the addresses that follow the address provided here.

MSB LSB

7	6	5	4	3	2	1	0
Address Byte							

If the data payload is greater than one, the interface is used in burst mode with automatic address increment.

9.10.3 N Bytes: data payload

The <Data Payload> field has a size (N) and can vary from one byte to 128 bytes. It contains the data that is to be written to or read from the target register(s).

MSB LSB

7	6	5	4	3	2	1	0
Data Payload							
...							
Data Payload							

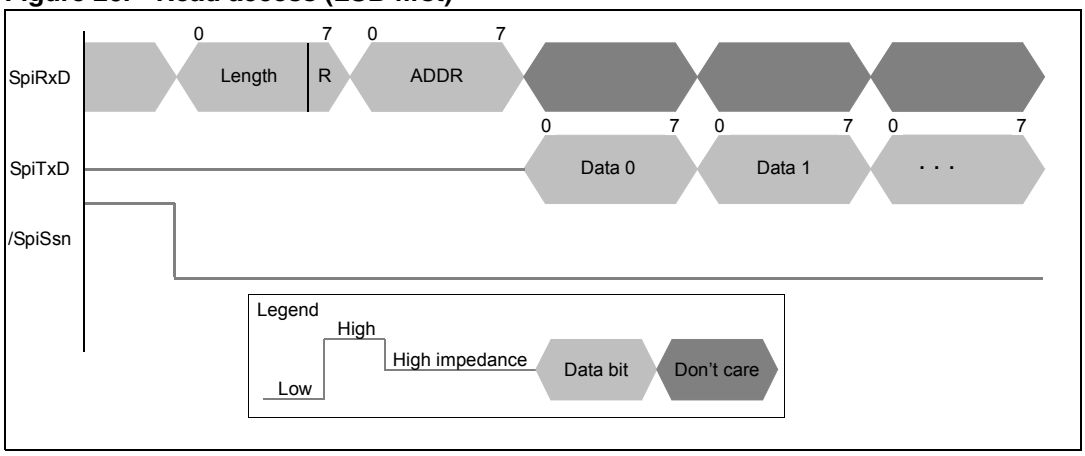
9.11 SPI read operations

To read data from the TN100 transceiver, do the following:

1. In the <Instruction> field, set the direction bit to the value 0 (read data) and set the data size, where the maximum size is 128.
2. In the <Address> field, set the source address.
3. Send the <Instruction> and <Address> data to the slave device(s).
4. Read in the resultant data from the slave device(s).

Figure 26 shows the timing of a read operation.

Figure 26. Read access (LSB first)



Note: The SpiTxD pin is activated into push-pull or open-drain mode only if the data payload bytes are transmitted and if the slave device is selected through SpiSsn. Otherwise, the SpiTxD pin is always in the high impedance state.

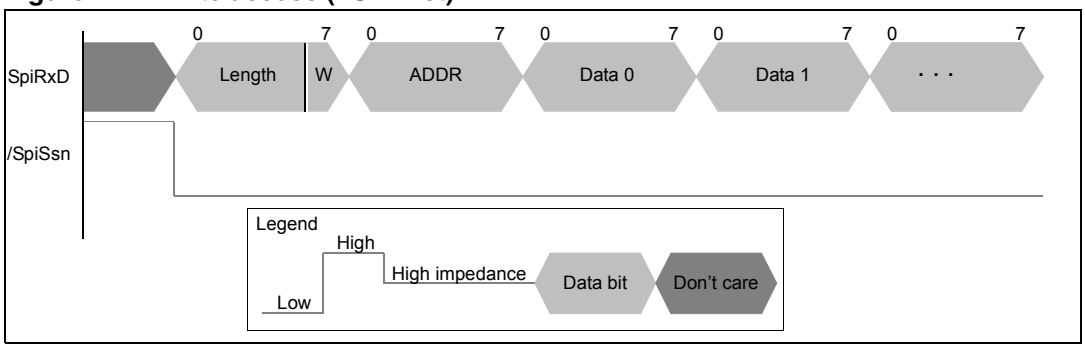
9.12 SPI write operations

To write data to the TN100 transceiver, do the following:

1. In the <Instruction> field, set the direction bit to the value 1 (write data) and set the data size, where the maximum size is 128.
2. In the <Address> field, set destination address.
3. In the <Data Payload> field, set the data that will be written to the slave device(s).
4. Send the <Instruction>, <Address>, and <Data Payload> data to the slave device(s).

The timing of a write operation is shown in the following figure:

Figure 27. Write access (LSB first)



Note: During a write operation the `SpiTxD` pin is always in the high impedance state.

9.12.1 Write access example

The following shows a write access example where three bytes (0x11, 0x22 and 0x33) are written to three registers that start at the address location 0x23:

<0x83><0x23><0x11><0x22><0x33>

Instruction field | Address field | Data payload field

Instruction field

First, the chip needs to know if this request is a read or write command, and how many bytes of data will be written to the register(s). In this example, the `<Instruction>` field has a value of `<0x83>`, which has a binary value of 10000011, as shown below:

MSB LSB

7	6	5	4	3	2	1	0
1	0	0	0	0	0	1	1
Write Access	Number of Bytes = 3						

The bit direction is set to write with three bytes of data to follow that are to be written.

Address field

Next, the chip needs to know to where in the register these three bytes of data are to be written (that is, which offset address to point to). In this example, the `<Address>` field has a value of <0x23>, which has a binary value of 00100011, as shown below:

MSB LSB

7	6	5	4	3	2	1	0
0	0	1	0	0	0	1	0
Address location = 0x23							

The data bytes are be written beginning at the offset address 0x23.

N data fields (3 bytes)

Finally, include the data to be written beginning at the offset address 0x23. In this example, the <Data Payload> field has the values 0x11, 0x22, and 0x33, which have the binary values 00010001, 00100010, and 00110011, respectively, as shown in the following diagram:

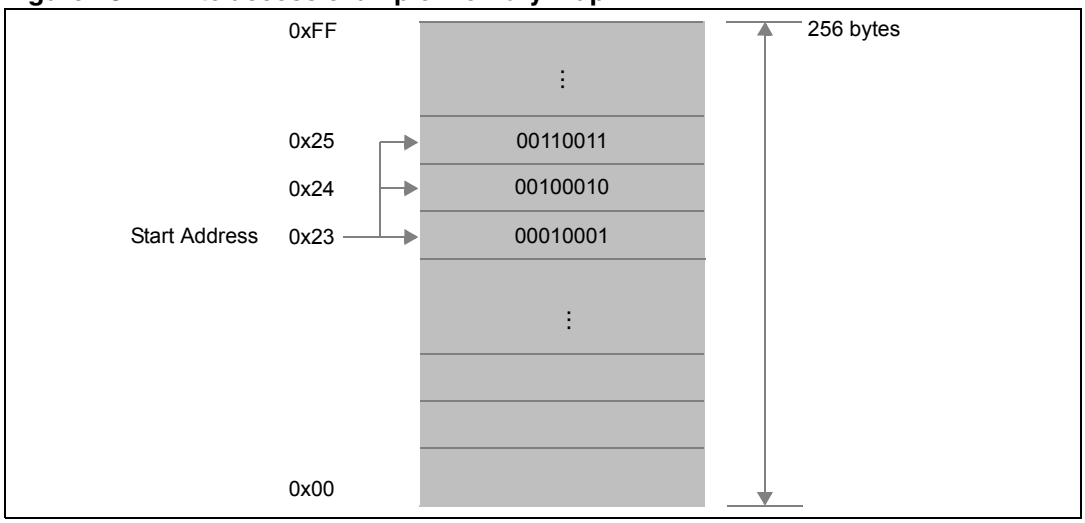
MSB LSB

7	6	5	4	3	2	1	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1

The three values 00010001, 00100010, and 00110011 will be written to the chip registers beginning at offset address 0x23, that is offset addresses 0x23, 0x24, and 0x25.

Figure 28 shows the location of the three registers in the chip register where the data provided has been written.

Figure 28. Write access example memory map



10 Chirp sequencer (CSQ)

This section describes in more detail the Chirp Sequencer RAM (CSQ) and how to program it, if required. Normally the CSQ is pre-programmed.

10.1 Purpose of the Chirp sequencer

The Chirp Sequencer (CSQ) memory space contains the values for I and Q, which are used to calculate Upchirps and Downchirps. The CSQ is set with a default matrix that has a symbol duration of 4 µs (4000 ns) and a 22 MHz bandwidth.

Note: This section is provided for information purposes only and can be skipped if the Chirp Sequencer will not be modified.

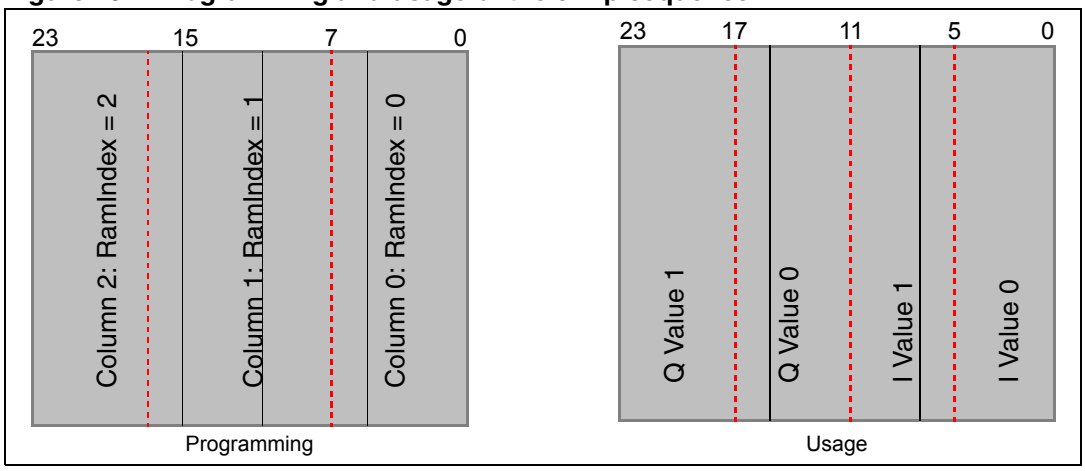
The output of the Chirp Sequencer delivers two sequences of 6-bit values that synthesize the I and Q signals of a symbol. The Chirp Sequencer uses a table of two 6 bit wide columns of 122 rows each. Both columns are read out simultaneously for each symbol that is transmitted. These signals are then processed as follows:

1. Converted into an analog signal.
2. Low pass filtered.
3. Modulated onto the carrier frequency.
4. Transmitted over the air.

The Chirp Sequencer table is programmable and must be initialized over the SPI interface by an initialization routine when the default matrix is not used. The use of a 6 bit value for each row in the table can cause additional overhead because 8 bit values are used in the SPI protocol and memory usually consist of bytes. Therefore, to avoid additional time for data transactions and unnecessary increased code size, the Chirp Sequencer memory is equipped with an 8 bit programming interface.

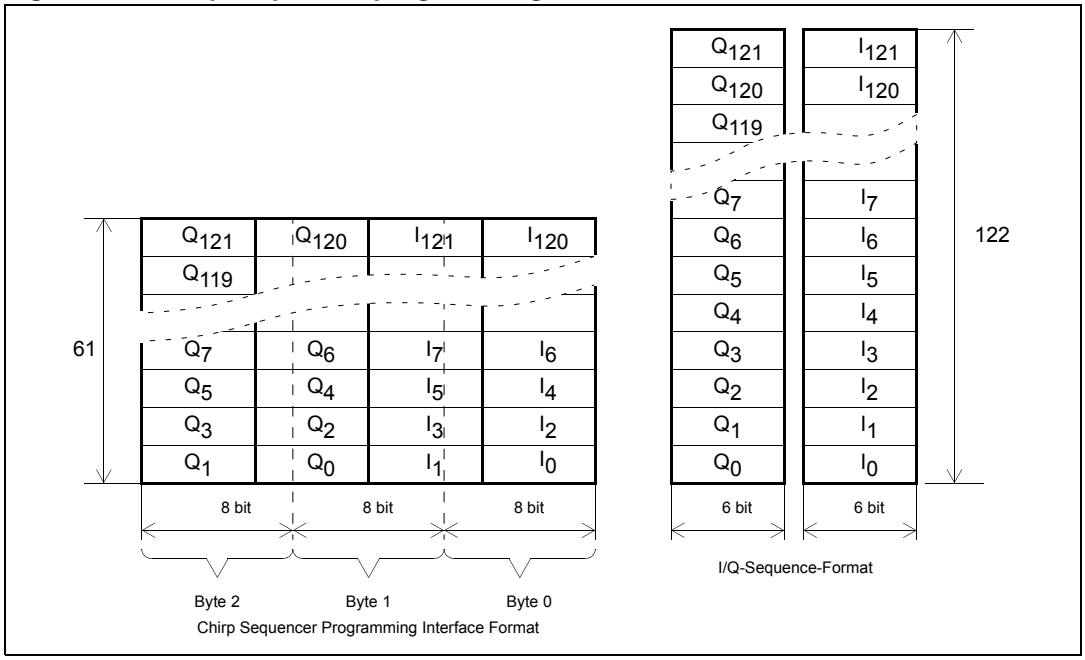
Three columns of 8 bits each are used for programming the CSQ, while four columns of 6 bits each are used by the TN100 transceiver to calculate the Upchirps and Downchirps. *Figure 29* illustrates these columns.

Figure 29. Programming and usage of the chirp sequencer RAM



On this programming interface, the Chirp Sequencer memory is translated into a table containing three columns of 8 bits each with each column having 61 rows (see *Figure 30*).

Figure 30. Chirp sequencer programming interface format



Usually the I and Q sequences are given in two lists of 6 bit values. Before the data from both lists can be written into the Chirp Sequencer memory, they must be formatted for the 8 bit programming interface. This can be done prior to the compilation because the I and Q sequences are not commonly changed during the lifetime a software version.

10.2 Formatting the I and Q values

The I and the Q values are formatted for the programming interface as follows:

1. Read in 122 elements of both lists (I and Q) into two separate arrays (I and Q array).
2. Allocate a two dimensional array of 3 columns by 61 rows.
3. Get the first and second I-value from the I-array.
4. Shift the second I-value 6 bits to the left, while discarding the upper 4 bits and merge it with the first I-value to an 8 bit value.
5. Write this 8 bit value to the first row of the least significant column (byte 0) of the two dimensional array.
6. Proceed this way (repeat steps 3 to 5) with the next I-values until the I-array is exhausted and the byte-0-column is filled up.
7. Get the second I-value from the I-array and the first Q-value from the Q-array.
8. Shift the second I-value 2 bits to the right, while discarding the lower 2 bits, shift the first Q-value 4 bits to the left, while discarding the higher 2 bits and merge both to an 8 bit value.
9. Write this 8 bit value to the first row of the middle column (byte 1) of the two dimensional array.

10. Proceed this way (repeat steps 7 to 9) with the next odd I-value and the next even Q-value until both arrays are exhausted and the byte-1-column is filled up.
11. Get the first and second Q-value from the Q-array.
12. Shift the first Q-value 4 bits to the right while discarding the lower 4 bits, and then shift the second Q-value 2 bits to the left and merge both to an 8 bit value.
13. Write this 8 bit value to the first row of the most significant column (byte 2) of the two dimensional array.
14. Proceed this way (repeat steps 11 to 13) with the next Q-values until the Q-array is exhausted and the byte-2-column is filled up.

10.3 CSQ writing example

Note: Only first row of CSQ writing is depicted.

```
I1 = 11.1111b
I2 = 10.0111b
Q1 = 10.1000b
Q2 = 10.0001b
+  : Means operation or

4.  I1       = 0011.1111b       = 0011.1111b
    I2 << 6 = 0010.0111b << 6 = 1100.0000b +
-------------------------------------------
row[0]column[0]                 = 1111.1111b
8.  I2 >> 2 = 0010.0111b >> 2 = 0000.1001b
    Q1 << 4 = 0010.1000b << 4 = 1000.0000b +
-------------------------------------------
row[0]column[1]                 = 1000.1000b
12. Q1 >> 4 = 0010.1000b >> 4 = 0000.0010b
    Q2 << 2 = 0010.0001b << 2 = 1000.0100b +
-------------------------------------------
row[0]column[2]                 = 1000.0110b
```

11 Packet and MACFrames

This section describes TN100 transceiver packets, including general packet types and the Preamble, MACFrame, and Tail fields in a packet.

11.1 General packet format

Each packet consists of four frame fields: a Preamble, a SyncWord, a MACFrame, and a Tail.

Figure 31. General packet format

Note: All bits in the fields are transmitted with the LSB first.

The bit and symbol size of each of these fields are described in *Table 25* for the 2ary modulation system, which is one bit per symbol.

Table 25. 2ary packet format

Field	Bits	Symbols
Preamble	30	30
SyncWord	64	64
MACFrame	80 to 114,996	80 to 114,996
Tail	–	4

11.2 General packet types

All packet types require the Preamble, the SyncWord and the Tail fields. *Table 26* lists the six different packet types defined in a TN100 transceiver system:

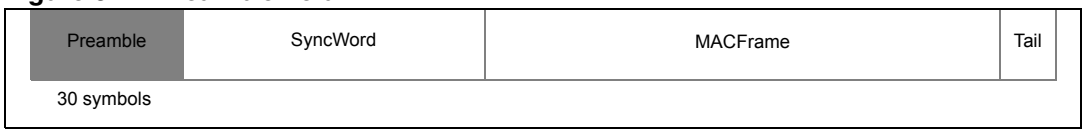
Table 26. TN100 defined packet types

Packet Type	Name	Description
Data	Data	The MACFrame of a Data packet can contain up to 8192 data bytes.
Acknowledgement	Ack	Acknowledges the successful reception of a Data packet.
Broadcast	Brdcast	Give specific information to either all stations in range (a broadcast message) or only a group of stations in range (a multicast message).
Time Beacon	TimeB	Gives all stations in range time information.
Request to Send	Req2S	Requests a frame transmission and reserves bandwidth by a station in a randomly accessed medium (CSMA/CA).
Clear to Send	Clr2S	Confirms a requested frame transmission of a station through a Req2S in a randomly accessed medium (CSMA/CA) and indicates that a Data packet can be transmitted.

11.3 Preamble field

The Preamble field is a fixed-length sequence of length `MacPreambleSymbols` (30) symbols in the 2ary modulation system and is used to facilitate AGC calibration and bit synchronization.

Figure 32. Preamble field



Note: *`MacPreambleSymbols` is defined in Appendix A: Attributes and constants on page 221.*

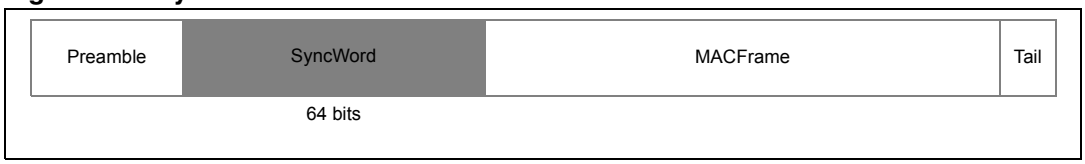
The sequences of symbols in the Preamble should use following modulation system:

- Upchirp/Downchirp – an alternating sequence of Upchirp/Downchirp pulses.

11.4 SyncWord field

The SyncWord entity is a 64 bit code word derived from a 24-bit logical network ID (LNID) and is used for frame synchronization.

Figure 33. SyncWord field



Different SyncWords can be used to facilitate frame synchronization on packets from different stations, networks, or logical channels. They are based on a (64,30) expurgated block code with the PN-overlay 0x03848D96BBCC54FC to get good auto correlation properties. A Hamming distance of 14 is guaranteed between SyncWords based on different LNIDs to allow frame synchronization with the presence of bit errors.

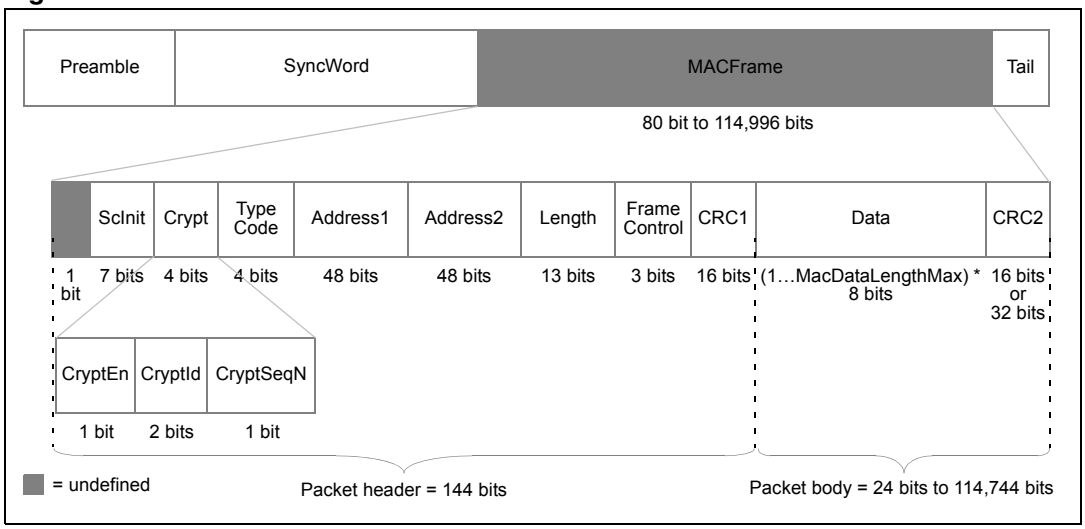
The following steps describe how the SyncWord is generated:

1. The information sequence is generated by appending 6 bits to the 24 bit LNID. If the MSB of the LNID equals the value “0”, then the appended bits are 010101. If the MSB of the LNID is the value “1”, then the appended bits are 101010.
2. The information sequence is pre-scrambled by XORing it with the bits 34 to 63 of the *pseudo-random noise* (PN) sequence (0x03848D96BBCC54FC).
3. The parity bits are created through polynomial division of the information sequence by the generator polynomial 260534236651 (in octal notation). Parity bits are the rest of the division.

11.5 MACFrame field

The contents of the MACFrame field defines the packet type. The size of the MACFrame can range from 80 bits for an Ack packet to 114,996 bits for a Data packet using FEC. The MACFrame field can consist of up to 10 fields depending on the packet type, as shown in *Figure 34*.

Figure 34. MACFrame field



Note: FEC can be performed over all fields of the MACFrame.

The following provides a detailed description of all possible fields in a MACFrame entity.

ScrInit (ScramblerInit)

The ScramblerInit (ScrInit) field is used to initialize the bit scrambler at the beginning of the receive cycle. The transmitter must set this field with the initial value of the scrambler since this value is not known at the receiver. This value can be changed for every packet.

The receiver uses the ScramblerInit value for descrambling the MACFrame. Scrambling is performed over all fields of the MACFrame (except the ScramblerInit field) to randomize the data using highly redundant patterns so that the probability of large zero patterns is minimized.

Crypt (EncryptionControl)

The EncryptionControl (Crypt) field is 4 bits in length and consists of the encryption enable switch (CryptEn), the encryption ID (CryptID), and the sequential numbering scheme (CryptSeqN) for encryption.

The CryptEn bit is used to indicate that encryption is used.

The 2 bit CryptID field, which is used for encrypted Data, Brdcast and TimeB packets, selects one of four secret keys (CryptKey, CryptClock) which are found in page one of RAM.

The CryptSeqN bit, which always starts with zero, increments CryptClock if the current CryptSeqN bit is unequal to the previous bit.

TypeCode

The TypeCode field is 4 bits in length. It identifies the packet type. Possible values include Data, Ack, Brdcast, TimeB, Req2S, and Clr2S.

A device that receives a frame with an invalid TypeCode will discard the packet without indication to the sending station or to the LLC.

Address1

The Address1 field is 48 bits in length and contains the destination address in all packet types, with the exception of the TimeB packet where the Address1 field contains a real-time clock value.

Address2

The Address2 field is 48 bits in length and contains the source address in Data, Brdcast, TimeB and Req2S packet types.

Length

The Length field is 13 bits in length and specifies the number of bytes of the Data field. It exists only in the Data, Brdcast, TimeB, Req2S and Clr2S packet types.

All bits set to “0” indicate a Data field of 8192 bytes. The maximum Length value allowed in a subset of this specification is defined by the MacDataLengthMax attribute. A station receiving a MACFrame with a value greater than the MacDataLengthMax (Data packet) or the MacBrdCastLengthMax (Brdcast packet) discards the packet without indication to the sender or to the LLC.

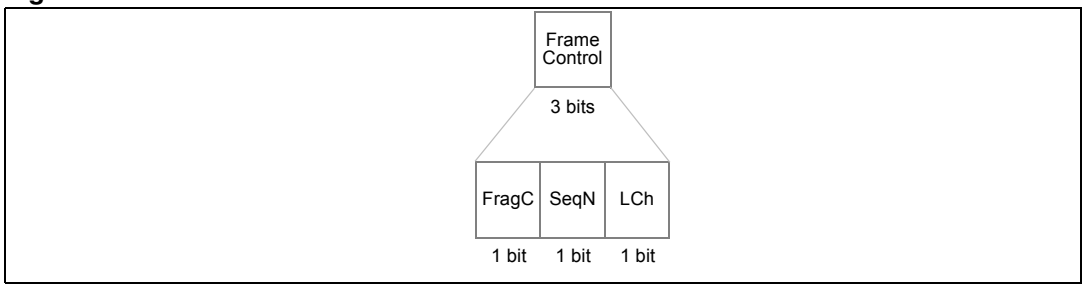
Note: MacDataLengthMax and MacBrdCastLengthMax are defined in Appendix A: Attributes and constants on page 221.

FrameControl

The FrameControl field is 3 bits in length and is transparent for the TN100 transceiver. For TX, these bits are loaded from RAM together with the TxLength field and sent in the MACFrame. For RX, these bits are stored in RAM together with the RxLength field received in the MACFrame where they can be read and used by an application.

In several applications the FrameControl field is used on the software level to implement functions such as Link Control Channel (where the third bit is defined as a logical channel bit LCh), a sequential numbering scheme (where the second bit is defined as a sequence numbering bit SeqN), and/or a packet fragmentation scheme (where the first bit is defined as a fragmentation control bit FragC). These bits are as shown in *Figure 35*.

Figure 35. FrameControl field used for Network Protocol



CRC1

The CRC1 field is 16 bits in length and performs a Cyclic Redundancy Check over the MACFrame header information. The CRC1 allows a checking of the integrity of the protocol information (such as addresses and length) in the MACFrame so that the MACFrame header information is immediately verified and is independent of the CRC mode of the data packet.

The CRC1 is calculated on the following MACFrame fields: EncryptionControl, TypeCode, Address1, Address2, and Length.

Data

The Data field is a variable length field that contains the data for the logical channels. It exists only in the Data and Brdcast packets. The Length field specifies the length in bytes of the Data field.

The minimum length is one byte. The maximum length is defined by the MacDataLengthMax and MacBrdcastLengthMax attributes, which is 8192 bytes.

Note: MacDataLengthMax and MacBrdCastLengthMax are defined in Appendix A: Attributes and constants on page 221.

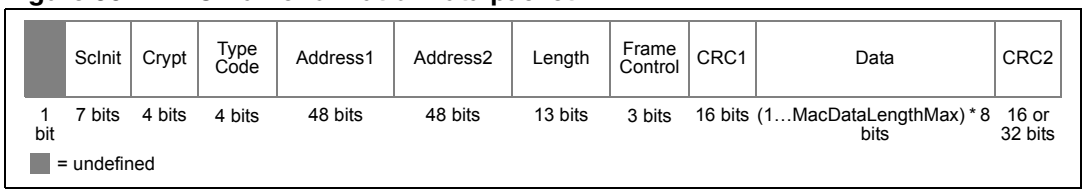
CRC2

The CRC2 field is 16 or 32 bits in length and performs a cyclic redundancy check on the Data field of Data and Brdcast packets. Different polynomials of 16 or 32 bit lengths are defined and can be selected.

11.5.1 MACFrame of a data packet

The MACFrame of a Data packet contains up to 8192 data bytes. A Data packet MACFrame is shown in *Figure 36*.

Figure 36. MACFrame format of Data packet



The following fields have information that is specific to the Data packet:

TypeCode Contains the type code of the Data packet.

Address1 Contains the destination address (DestinationAddress).

Address2 Contains the source address (SourceAddress).

Length Contains the length of the Data packet in bytes.

FrameControl These bits are transparent to the TN100 transceiver. They are used in *Ranging Application Software*, as follows:

Table 27. Ranging Data Packets

Ranging Data Packet Type	Value
Normal Data Packet	0x00
Ranging_Start Packet	0x01
Ranging_Answer1 Packet	0x02
Ranging_Answer2 Packet	0x03
Ranging_Fast_Start Packet	0x04
Ranging_Fast_Answer1	0x05

Note: FragC bit (bit 1) = LSB. LCh bit (bit 3) = MSB.

Data

The maximum length of the Data field used in an implementation subset of this specification is defined by the MacDataLengthMax attribute.

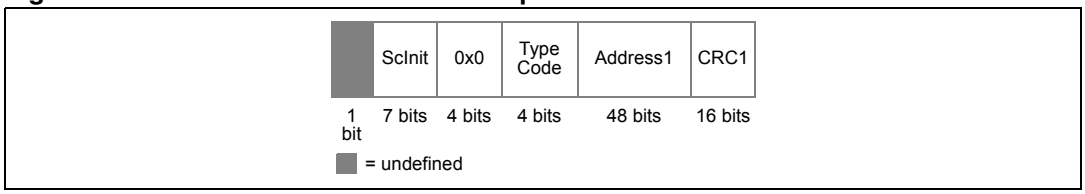
If messages with more than MacDataLengthMax bytes have to be transmitted, they can be fragmented into several Data packets of different lengths. If a Data packet is received with more than MACDataLenghtMax bytes, it is discarded without indication to the sender or to the LLC.

Note: MacDataLengthMax is defined in Appendix A: Attributes and constants on page 221.

11.5.2 MACFrame of an acknowledgement (Ack) Packet

An Ack packet is sent by the receiver to acknowledge the successful reception of a Data packet. The MACFrame of an Ack packet is shown in *Figure 37*.

Figure 37. MACFrame format of an Ack packet



The following fields have information that is specific to the Ack packet:

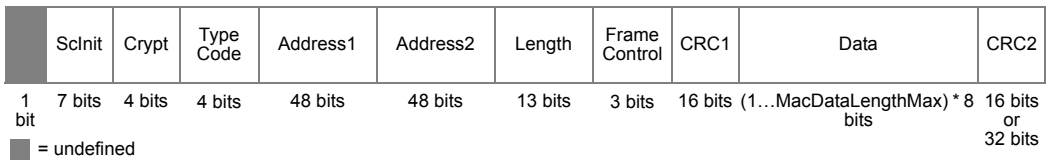
TypeCode Contains the type code of the Ack packet.

Address1 Contains the destination address (DestinationAddress).

11.5.3 MACFrame of a broadcast (Brdcast) packet

A Brdcast packet is sent by a station to give specific information to either all stations in range (a broadcast message) or only a group of stations in range (a multicast message). The MACFrame of a Brdcast packet is shown in *Figure 38*.

Figure 38. MACFrame format of a Brdcast packet



The following fields have information that is specific to the Brdcast packet:

FrameControl

Since a Brdcast packet is sent as a broadcast to more than one station, it must not be acknowledged by any station receiving the packet. Because of the lack of retransmission possibilities for Brdcast packets (unacknowledged transmission) fragmentation is impossible. Therefore, the FragC and SeqN bits must be set to the value “0”.

Data

The maximum length of the Data field used in a implementation subset of this specification is defined by MacBrdCastLengthMax attribute. If a Brdcast packet is

received with more than MACBrdCastLengthMax bytes, it will be discarded without indication to the LLC.

Note: MacBrdCastLengthMax is defined in Appendix A: Attributes and constants on page 221.

TypeCode

Contains the type code of the Brdcast packet.

Address1

Contains either a broadcast address (BroadcastAddress) or a multicast address (MulticastAddress).

Address2

Contains the source address (SourceAddress).

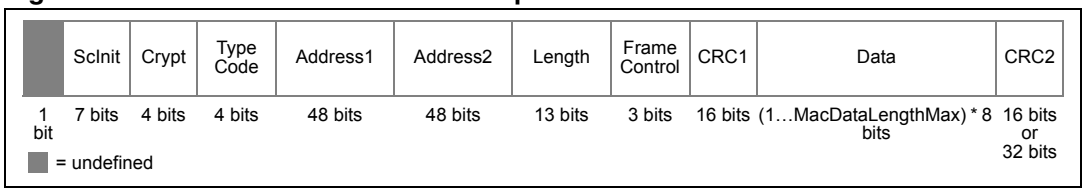
Length

Contains the length of the packet in bytes.

11.5.4 Time beacon packet (TimeB) MACFrame

A TimeB packet is sent by a station to give all stations within range time information. This packet allows synchronization of the real-time clocks of stations. Since the time beacon is sent as a broadcast to more than one station, it must not be acknowledged by any station receiving it. A TimeB packet MACFrame is shown in *Figure 39*.

Figure 39. MACFrame format of TimeB packet



The following fields have information that is specific to the TimeB packet:

Address1

The Address1 field carries a 48 bit real-time clock value (RTCvalue).

The transmitting station must adjust the RTCvalue before sending it over the physical interface (antenna). When the last bit (MSB) of the RTCvalue appears on the antenna, the value must be identical to the real-time clock value of the transmitting station. This is required to compensate for transmitter delay.

Additionally, the receiving station must adjust the RTCvalue before updating its real-time clock. When the TimeB packet has been processed, the RTCvalue must be adjusted to compensate for receiver delay.

Address2

The Address2 field contains the source address (SourceAddress) of the TimeB packet.

Data

The Data field of the TimeB packet can be used to transmit a signature to authenticate the RTCvalue and the source address of the TimeB packet or to broadcast specific information.

TypeCode

Contains the type code of the TimeB packet.

Length

Contains the length of the packet in bytes.

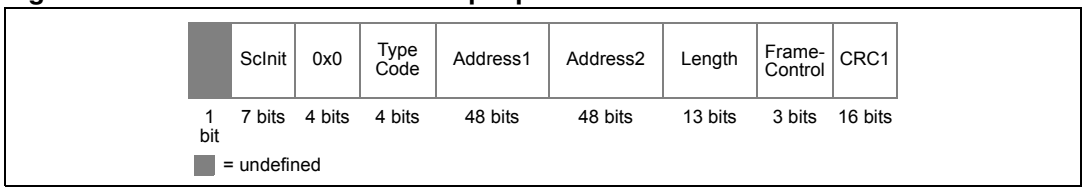
FrameControl

Only the LCh bit is defined.

11.5.5 Request to send (Req2S) packet MACFrame

The Req2S packet requests a frame transmission and reserves bandwidth by a station. The requested station responds with a Clear to Send packet (Clr2S) to the requesting station. A Req2S packet MACFrame is shown in *Figure 40*.

Figure 40. MACFrame format of Req2S packet



The following fields have information that is specific to the Req2S packet:

Length

The Length field indicates the length of the Data packet (in bytes) of the requested transmission.

TypeCode

Type code of the Req2S packet.

Address1

Contains the destination address (DestinationAddress).

Address2

Contains the source address (SourceAddress).

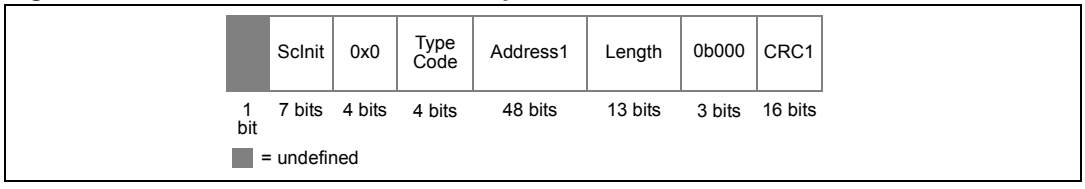
FrameControl

These fields are undefined.

11.5.6 Clear to send (Clr2S) packet MACFrame

The Clr2S packet confirms a requested frame transmission of a station through a Req2S packet, indicating that a Data packet can be transmitted. The Addess2 and Length fields are copied from the Req2S packet and sent back to the requesting station. A Clr2S packet MACFrame is shown in *Figure 41*.

Figure 41. MACFrame format of Clr2S packet



The following fields have information that is specific to the Clr2s packet:

TypeCode

Contains the type code of the Clr2s packet.

Address1

Contains the destination address (DestinationAddress).

Length

Contains the length of the packet in bytes.

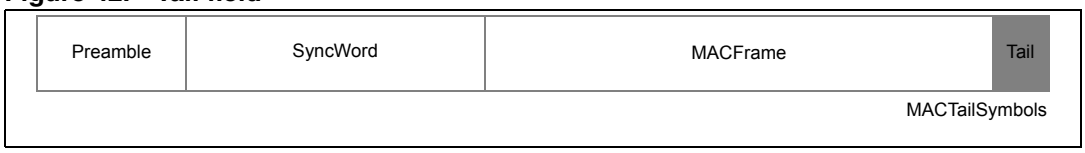
FrameControl

These fields are undefined.

11.6 Tail field

The Tail field is used as a delimiter between the end of a packet. Tail symbols are required in on-off-keying systems for the physical carrier sensing mechanism. For instance, consecutive off-symbols at the end of a packet and the following interframe space interval could be misinterpreted by stations trying to access the media as a physically idle medium.

Figure 42. Tail field



The Tail is a fixed-length sequence of MacTailSymbols (4) bits defined below for the modulation systems:

- Upchirp/Downchirp – an alternating sequence of Upchirp and Downchirp pulses.
- Upchirp/Off – a continuous sequence of Upchirp pulses.
- Downchirp/Off – a continuous sequence of Downchirp pulses.

12 Frame control scheme

This section describes a frame control scheme that can be implemented in software. This schemes use the FrameControl field of the MACFrame, which is transparent to the TN100 transceiver.

12.1 Logical channels

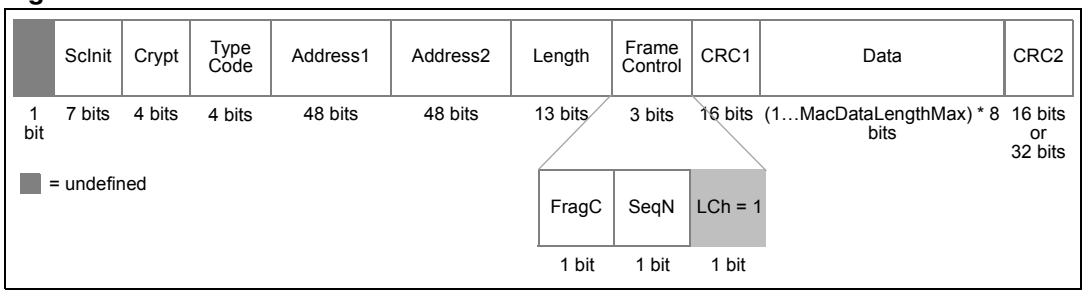
The Link Control Manger should be implemented in the Data Link Layer. The following describes the settings of the LCh bit in the FrameControl field which selects between the Link Control Channel and the User Data Channel.

12.1.1 Link control channel

The LCh bit is used to distinguish between the two logical channels (LCCh and UDCh). A value “0” indicates the user data channel (UDCh) while a value “1” indicates the link control channel (LCCh).

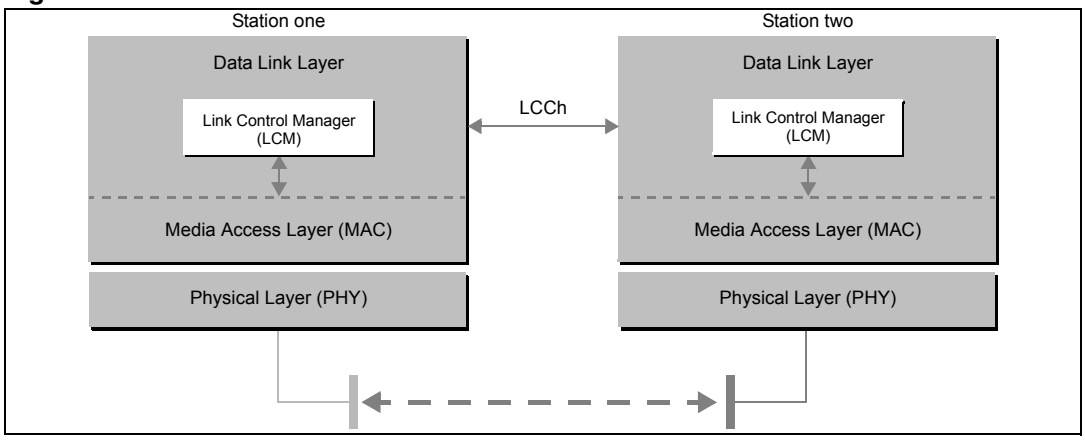
The Link Control Channel (LCCh) carries control information that is exchanged between the Link Control Managers (LCM) of two stations. The LCCh is indicated by the Logical Channel bit (LCh) set to the value “1” in the FrameControl field of Data or Brdcast packets.

Figure 43. LCh field set to LCCh



Information exchanged between two stations via the Link Control Channel is shown in *Figure 44*.

Figure 44. Link control channel

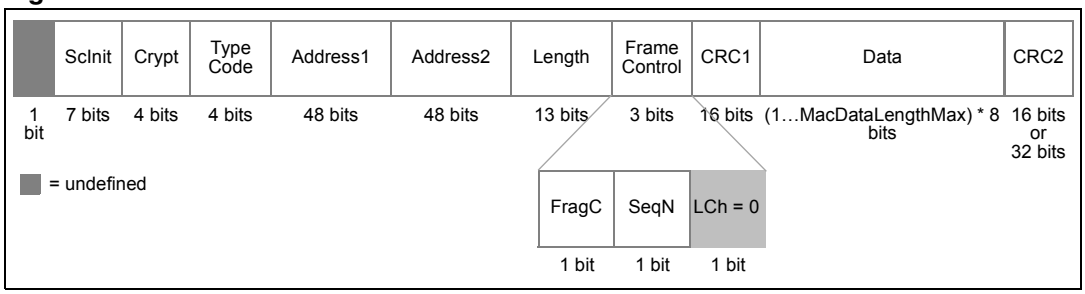


Note: For more details, see Section 12.2: Link control management on page 66.

12.1.2 User data channel

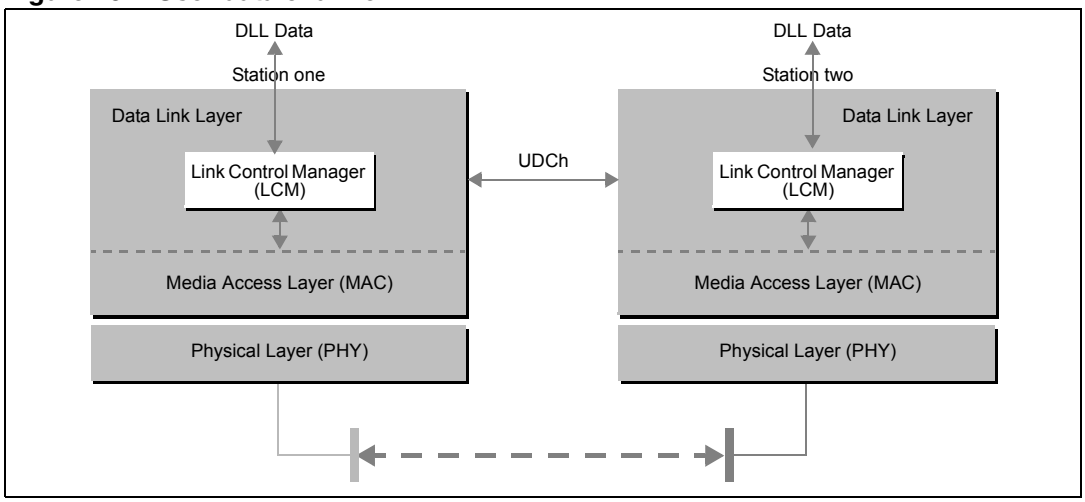
The User Data Channel (UDCh) carries user data that is exchanged between the Data Link Controls (DLC) of two stations. The UDCh channel is indicated by the Logical Channel bit (LCh) set to the value “0” in the FrameControl field of Data or Broadcast packets.

Figure 45. LCh field set to UDCh



Information exchanged between two stations via the User Data Channel is shown in *Figure 46*.

Figure 46. User data channel



12.2 Link control management

Link Control Management Protocol (LCMP) messages are used for link set-up, security, and control. They are transferred in the Data field instead of user data and are indicated by a bit with the value “1” in the LCh of the FrameControl field. The messages are filtered out and interpreted by the Link Control Management (LCM) on the receiving side and are not propagated to higher layers.

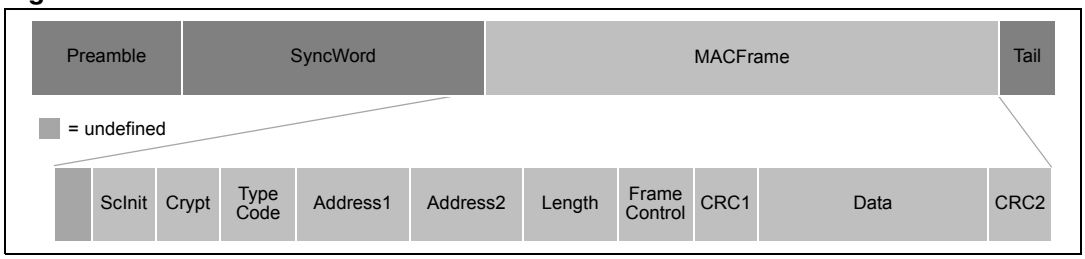
LCMP messages have higher priority than user data. This means that if the LCM needs to send a message, it is not delayed by user data traffic, although it can be delayed by many retransmissions of individual packets.

13 MACFrame configuration (Auto/Transparent)

This section explains how to configure the MACFrame in Auto and Transparent mode.

A transmitted or received packet contains a Preamble, a SyncWord, the MACFrame, and a Tail, which are used by the TN100 transceiver to help process the received packet. The MACFrame, however, contains the raw data which is used by upper layers.

Figure 47. MACFrame contents



The TN100 transceiver provides a wide range of features that can be automatically set in the MACFrame as it is prepared for transmission, or read out when a MACFrame is received. These include Forward Error Correction, Encryption, Address Matching, and so on. However, the chip provides the option of allowing software to configure the MACFrame. These two modes are *Auto Mode* and *Transparent Mode*:

- In Auto mode, for TX, the MACFrame is configured by automatically by hardware (the TN100 transceiver) using data provided in chip register settings and buffers, while for RX, the hardware (the TN100 transceiver) parses the received MACFrame and stores the resultant values in chip registers and buffers for further use by software.
- In Transparent mode, for TX, software configures and prepares the MACFrame and stores in in a buffer for transmission, while for RX, the MACFrame is “transparent” to the chip and stores the entire MACFrame in a buffer for further use by software.

13.1 Field for setting MACFrame auto and transparent mode

The following field is used for setting MACFrame Auto and Transparent Mode.

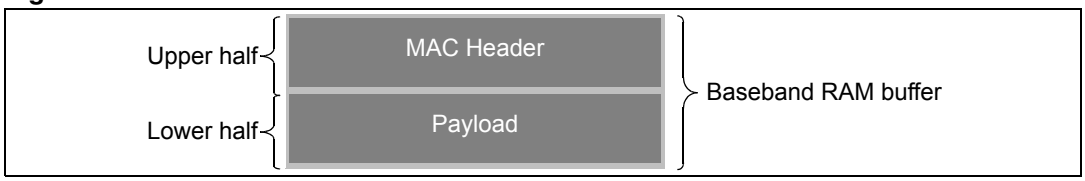
Table 28. Wake-up time fields

Field	Offset	R/W	Description
TxRxBbBufferMode0	0x4A	WO brst BbClk	Sets the TX and RX data as either two segments (Auto) or all four segments (Transparent).

13.2 MACFrame auto mode

When the TN100 transceiver is set to Auto Mode, the MAC header values and data payload used for TX and RX are stored in a single buffer which has been divided into an upper and a lower part. The upper half stores MAC header data while the lower half stores the packet payload.

Figure 48. Auto mode



For TX, the digital part of the chip builds the MAC header using values stored in chip buffers and registers, including the payload data. Additional packet requirements (Preamble, SyncWord and Tail) are added to this completed MACFrame and the packet is transmitted.

For RX, the Preamble, SyncWord and Tail are removed and the MAC header and payload of extracted MACFrame are stored in chip registers and buffers. Software can then read out the resultant MAC header values and payload data.

13.2.1 Setting MACFrame auto mode (default)

To set MACFrame Auto mode, do the following:

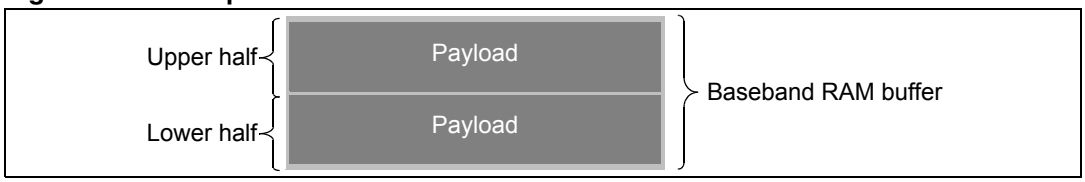
- Set `TxRxBbBufferMode0 = 0x0`

Note: Auto mode is the default setting (0x0)

13.3 MACFrame transparent mode

When the TN100 transceiver is set to Transparent Mode, MAC header values and data payload are prepared in software and a complete MACFrame must be provided to the chip. A single buffer is used to store the MACFrame, where both the upper half and the lower half are available for storing the complete MACFrame.

Figure 49. Transparent mode



For TX, software builds the MACFrame by assembling the MAC header and payload data. This assembled MACFrame is placed in the a baseband RAM buffer(s). The TN100 transceiver builds the packet by using this MACFrame and additional packet requirements (Preamble, SyncWord and Tail) after which the packet is transmitted.

For RX, the Preamble, SyncWord and Tail are removed and the MACFrame is extracted from the packet and stored in the baseband RAM buffer. This MACFrame is read out and provided to an application for further processing.

13.3.1 Setting MACFrame transparent mode

To set Transparent mode for the MACFrame, do the following:

- Set `TxRxBbBufferMode0 = 0x1`
  where `0x1 = NA_TxRxBufferMode0Transparent_BC_C`

14 Baseband RAM configuration

This section describes the four possible configurations of the baseband RAM using the TX/RX buffer and MACFrame modes, which are Auto/Duplex, Auto/Simplex, Transparent/Duplex, and Transparent/Simplex.

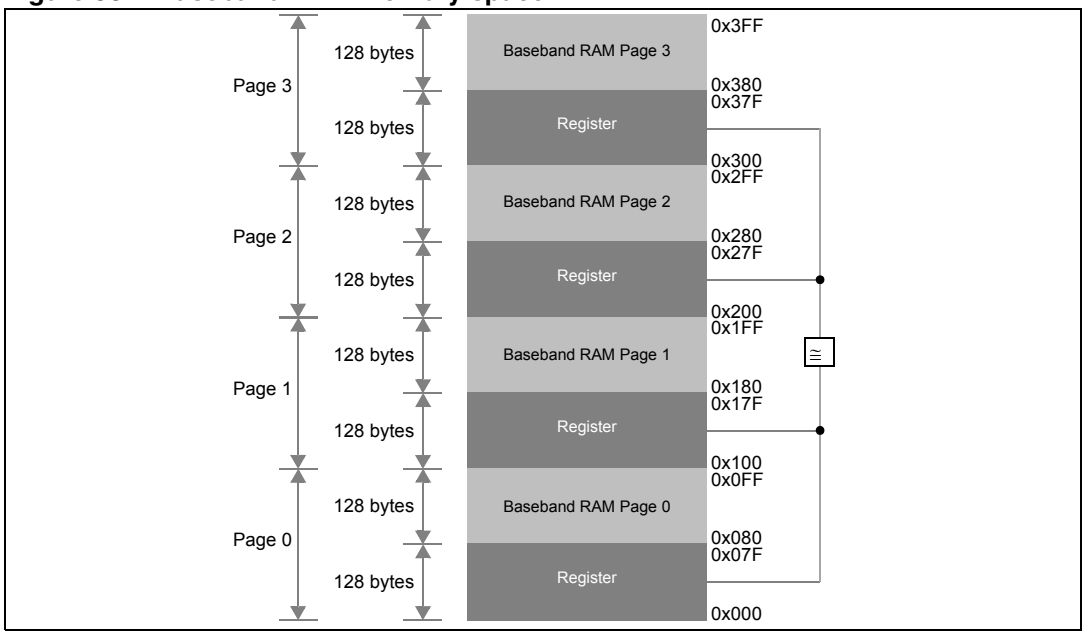
14.1 Overview

The 1024 byte memory space of the TN100 transceiver consists of four baseband RAM segments totaling 512 bytes.

Note: *See also Section 6: Memory map on page 23.*

Each of the four segments has a 128 byte register space, three of which are mappings of the chip register space (0x00 to 0x7F). The mappings of the register space and the four baseband RAM segments are shown in *Figure 50*.

Figure 50. Baseband RAM memory space



14.1.1 Configurable spaces

The chip register memory space is non-configurable. Values written to the register address spaces (0x00 - 0x7F, 0x100 - 0x17F, 0x200 - 0x27F and 0x300 - 0x37F) always access the same Register Block (e.g. 0x00 is equivalent to 0x100, 0x200 and 0x300).

Baseband RAM, however, is configurable. Pages 0 to 3 of the Baseband RAM is configurable to one of four mode combinations, as described in *Table 29*.

Table 29. Baseband RAM configuration modes

Mode	Description
Auto/Duplex Mode	Two separate data buffers of 128 bytes each of the baseband RAM are used (Page 2 and Page 3).
Auto/Simplex Mode	Two baseband RAM pages of 128 bytes each are combined for a single data buffer for both RX and TX (Page 2 and Page 3)
Transparent/Duplex	Two separate buffers of 256 bytes each are used for an RX data buffer and a TX data buffer (Page 0, Page 1, Page 2 and Page 3).
Transparent/Simplex	a single 512 byte RX/TX data buffer is used sequentially for either a RX data buffer or TX data buffer (Page 0, Page 1, Page 2 and Page 3).

14.2 Memory usage

Each combination of MACFrame configuration modes with Baseband buffer configuration modes results in different memory usages. *Table 30* shows the four possible combinations with the resultant memory usage:

Table 30. Baseband memory usage

Memory usage	Auto	Transparent
Simplex	256 bytes (1 X 256 bytes)	512 bytes (1 X 512 bytes)
Duplex	256 bytes (2 X 128 bytes)	512 bytes (2 X 256 bytes)

14.3 Auto/Duplex mode

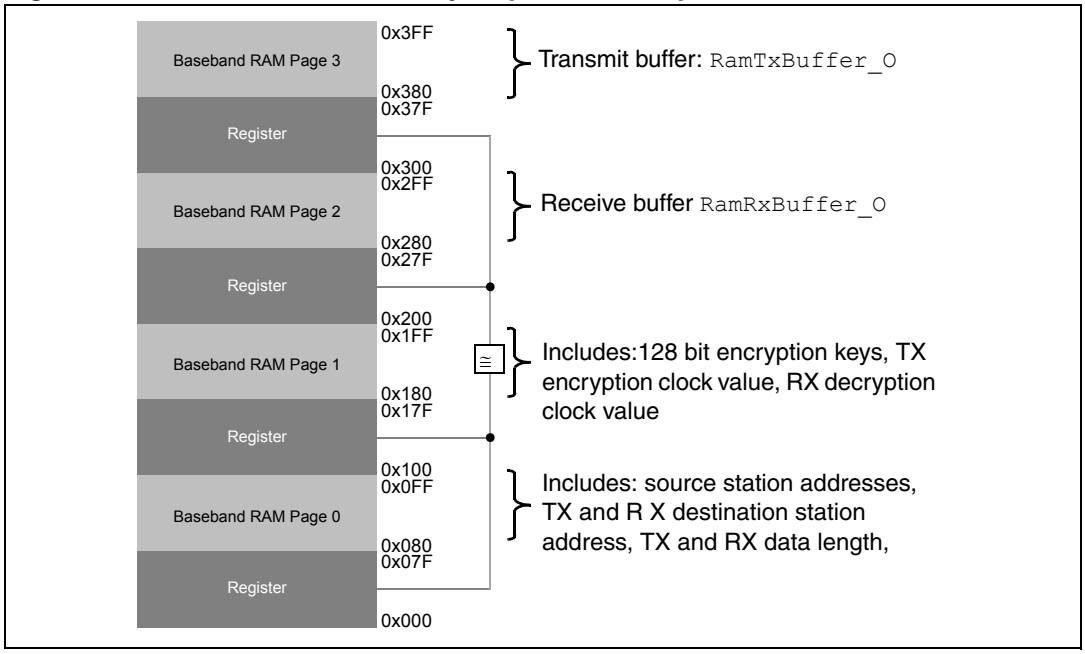
In Auto/Duplex mode, two separate data buffers of 128 bytes each of the baseband RAM are used. Only half of the baseband RAM is used for data (256 bytes in total) while the other half is used for MAC header values.

Table 31. Auto/Duplex mode configuration

Baseband RAM page	Size	Purpose
Page 0 and 1	256 bytes	Both pages are used for MAC header values.
Page 2	128 bytes	Single page used as a buffer for the data payload of received packets.
Page 3	128 bytes	Single page used for data payload of packets to be transmitted.

Figure 51 shows Auto/Duplex buffer configuration.

Figure 51. Baseband RAM memory map for Auto/Duplex mode



14.3.1 Setting Auto/Duplex mode

The following settings are made to configure the Baseband RAM for Auto / Duplex mode:

- `TxRxBbBufferMode0 = 0x0` (Auto)
- `TxRxBbBufferMode1 = 0x0` (Duplex)

14.3.2 Start addresses of the RX/TX data buffers

The RX data buffer (set by `RamRxBuffer_O`) consists of two blocks of 64 bytes each, where:

- Start address of lower block for RX = `RamRxBuffer_0`
- Start address of upper block for RX = `RamRxBuffer_0 + 0x040`

The TX data buffer (set by `RamTxBuffer_O`) consists of two blocks of 64 bytes each, where:

- Start address of lower block for TX = `RamTxBuffer_0`
- Start address for upper block for TX = `RamTxBuffer_0 + 0x040`

Note: See also Section 18: Buffer access synchronization on page 83.

14.4 Auto/Simplex mode

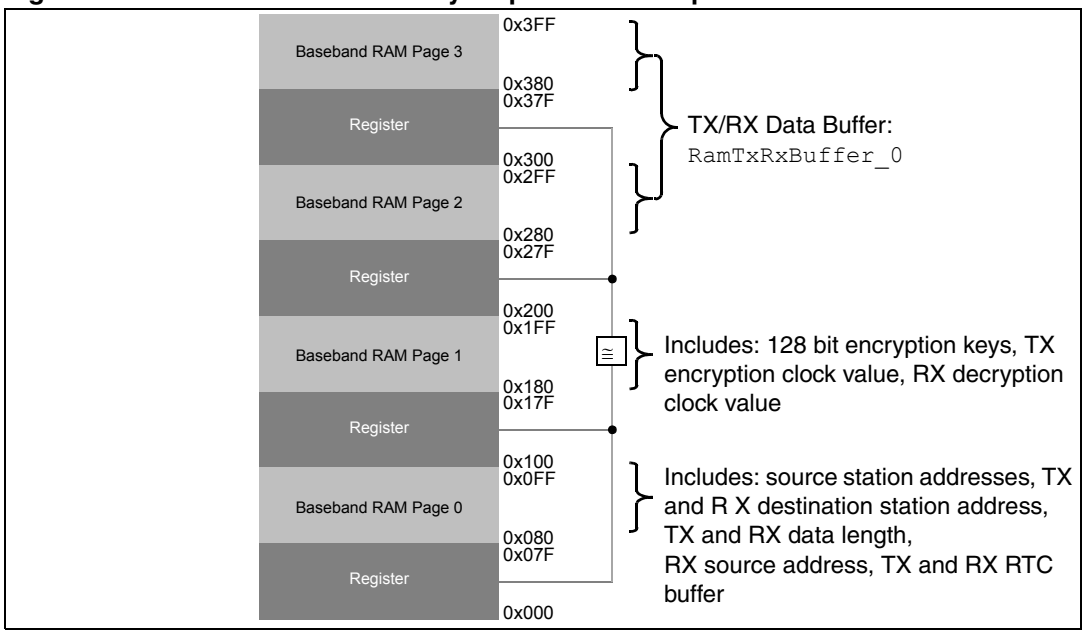
In Auto/Simplex mode, two baseband RAM pages of 128 bytes each are combined for a single data buffer for both RX and TX. Only half of the baseband RAM is used for data (256 bytes in total) while the other half is used for MAC header values.

Table 32. Auto/Simplex mode configuration

Baseband RAM page	Size	Purpose
Page 0 and 1	256 bytes	Both pages are used for MAC header values.
Page 2 and 3	256 bytes	Both pages are used as a data buffer for the data payload of received packets or packets to be transmitted.

Figure 52 shows a memory map of Auto/Simplex buffer configuration.

Figure 52. Baseband RAM memory map for Auto/Simplex mode



14.4.1 Setting Auto/Simplex mode

The following settings are made to configure the Baseband RAM for Auto/Simplex mode:

- `TxRxBbBufferMode0 = 0x0` (Auto)
- `TxRxBbBufferMode1 = 0x1` (Simplex)

14.4.2 Start addresses of the RX/TX data buffer

The TX / RX data buffer (set by `RamTxRxBuffer_0`) consists of two blocks of 128 bytes each, where:

- Start address of lower block for RX and TX = `RamTxRxBuffer_0`
- Start address of upper block for RX and TX = `RamTxRxBuffer_0 + 0x100`

Note: See also Section 18: Buffer access synchronization on page 83.

14.5 Transparent/Duplex mode

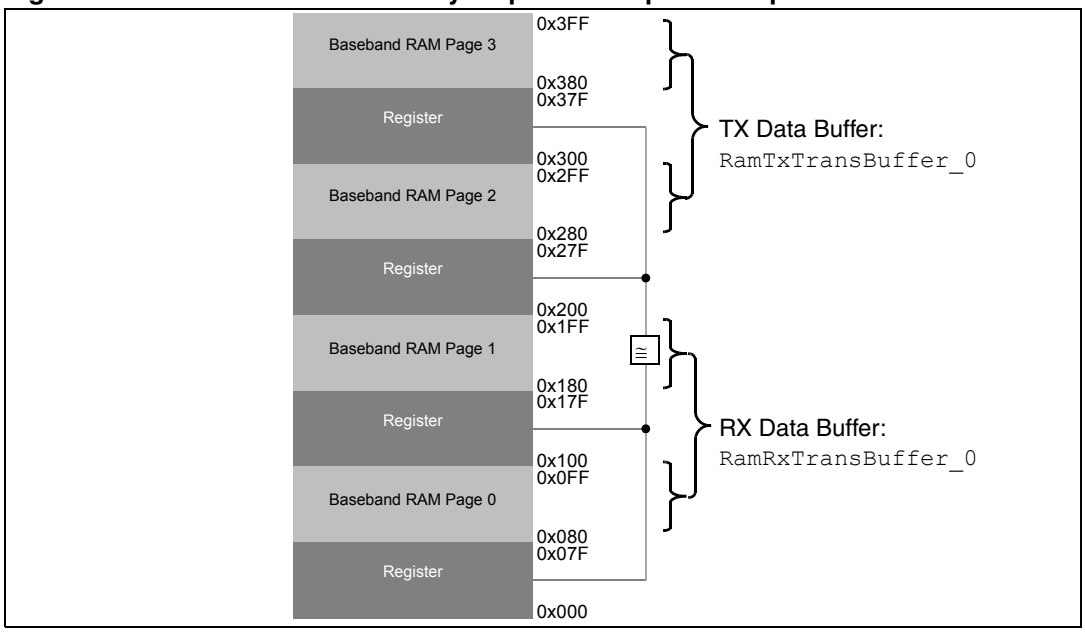
In Transparent/Duplex mode, two separate buffers of 256 bytes each are used for an RX data buffer and a TX data buffer. Software is responsible for configuring and reading the MACFrame, including MAC header values and data payload.

Table 33. Transparent/Duplex mode configuration

Baseband RAM page	Size	Purpose
Page 0 and 1	256 bytes	Both pages are used as a data buffer for received packets.
Page 2 and 3	256 bytes	Both pages are used as a data buffer for packets to be transmitted.

Figure 53 shows the memory map of Transparent/Duplex buffer configuration.

Figure 53. Baseband RAM memory map for Transparent/Duplex mode



14.5.1 Setting Transparent/Duplex mode

The following settings are made to configure the baseband RAM for Transparent/Duplex mode:

- `TxRxBbBufferMode0 = 0x1` (Transparent)
- `TxRxBbBufferMode1 = 0x0` (Duplex)

14.5.2 Start addresses of the RX/TX data buffers

The RX data buffer (set by `RamRxTransBuffer_0)` consists of two blocks of 128 bytes each, where:

- Start address of lower block for RX = `RamRxTransBuffer_0`
- Start address of upper block for RX = `RamRxTransBuffer_0 + 0x100`

The TX data buffer (set by `RamTxTransBuffer_0)` consists of two blocks of 128 bytes each, where

- Start address of lower block for TX = `RamTxTransBuffer_0`
- Start address of upper block for TX = `RamTxTransBuffer_0 + 0x100`

Note: See also Section 18: Buffer access synchronization on page 83.

14.6 Transparent/Simplex mode

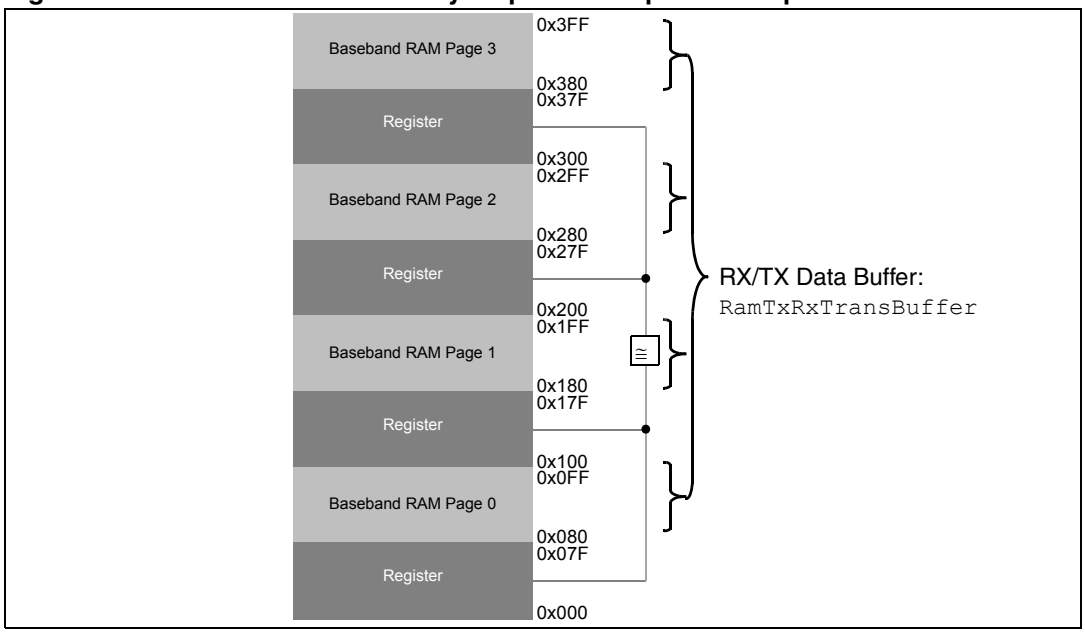
In this configuration, a single 512-byte RX/TX data buffer is used sequentially for either a RX data buffer or TX data buffer. It consists of the complete baseband RAM memory space of four 128 byte buffers. Software is responsible for configuring and reading the MACFrame, including MAC header values and data payload.

Table 34. Transparent/Simplex mode configuration

Baseband RAM Page	Size	Purpose
Page 0, 1, 2, and 3	256 bytes	All pages are used as a single data buffer for either received packets or packets to be transmitted.

Figure 54 shows the memory map of Transparent/Simplex buffer configuration.

Figure 54. Baseband RAM memory map for Transparent/Simplex mode



14.6.1 Setting Transparent/Simplex mode

The following settings are made to configure the baseband RAM for Transparent/Simplex mode:

- `TxRxBbBufferMode0 = 0x1` (Transparent)
- `TxRxBbBufferMode1 = 0x1` (Simplex)

14.6.2 Start addresses of the RX/TX data buffer

The RX/TX data buffer (set by `RamRxTransBuffer_0)` consists of two blocks of 256 bytes each, where:

- For the lower block:
  - Start address of lower half = `RamTxRxTransBuffer_0`
  - Start address of upper half = `RamTxRxTransBuffer_0 + 0x100`
- For the upper block:
  - Start address of lower half = `RamTxRxTransBuffer_0 + 0x200`
  - Start address of upper half = `RamTxRxTransBuffer_0 + 0x300`

Note: See also Section 18: Buffer access synchronization on page 83.

Warning: It is important to be aware that each of the two blocks are spread over two segments.

15 Buffer configuration (simplex/duplex)

This section describes how to configure the TX and RX buffers in Duplex and Simplex mode.

15.1 Overview

The TN100 transceiver can process packets for transmit and receive sequentially or simultaneously. In sequential mode or *Simplex Mode*, at any given time the chip can process either received packets or packets for transmission. In simultaneous mode or *Duplex Mode*, at any given time, the chip can process both received packets and packets for transmission. Each mode requires a configuration of the baseband RAM.

15.2 Field for setting duplex and simplex mode

The following field is used for setting Duplex or Simplex mode for the baseband TX/RX buffer.

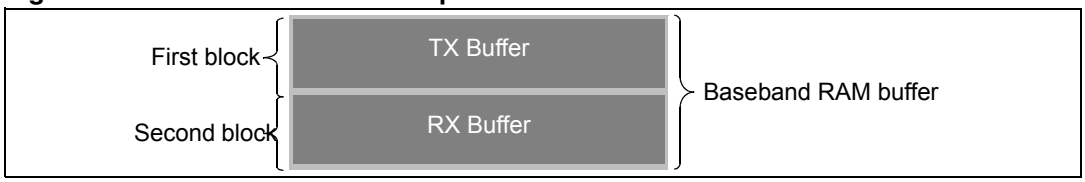
Table 35. Wake-up time fields

Field	Offset	R/W	Description
`TxRxBbBufferMode1`	0x4A	WO brst BbClk	Sets the data buffer for TX and RX as either two blocks (Duplex) or a single block (Simplex).

15.3 Duplex mode (default)

In Duplex mode, the baseband RAM buffer is split into two blocks, one for TX and one for RX. The TX and RX buffer operations are independent of each other. However, to allow simultaneous TX and RX buffer operations, smaller buffers are available for MACFrame data (MAC headers and data payload). The size of these blocks are dependent on the MACFrame mode (Auto or Transparent). Duplex mode is the default setting.

Figure 55. Baseband buffer in Duplex mode



Note: See Section 13: MACFrame configuration (Auto/Transparent) on page 67.

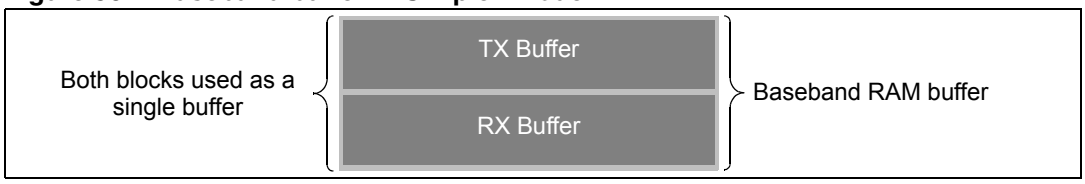
15.3.1 Configuring the baseband RAM for Duplex mode

To set the baseband RAM for Duplex Mode, set `TxRxBbBufferMode1 = 0x0`

15.4 Simplex mode

In Simplex mode, a single baseband RAM buffer of at least two pages is used for both TX and RX. This results in a larger buffer available for the MACFrames. The size of the buffer is dependent on the selection of MACFrame mode (Auto or Transparent).

Figure 56. Baseband buffer in Simplex mode



Note: See Section 13: MACFrame configuration (Auto/Transparent) on page 67.

15.4.1 Configuring the baseband RAM for Simplex mode

To set the baseband RAM for Simplex mode, set `TxRxBbBufferMode1 = 0x1`

16 Buffer swapping between TX and RX buffers

This section discusses the buffer swap feature of the TN100 transceiver, which exchanges the buffers between the baseband transmitter and receiver.

When enabled, data for the transmit operation are taken from the receive buffer, while data of the receive operation are stored in transmit buffer. Buffer swapping significantly eliminates the overhead of data transactions over the SPI interface and improves the decoupling between transmit or receive processes and data transactions. Also, when enabled, if a station is set up as an intermediate station (IS) for frame forwarding, the station does not have to move data from the RX buffer to the TX buffer.

Note: When buffer swapping is enabled, only Auto / Duplex mode and Transparent / Duplex mode can be enabled.

16.1 Enabling buffer swapping (duplex mode only)

When buffer swapping is enabled, data for the transmit operation is taken from the receive buffer, while data of the receive operation are stored in transmit buffer. The following field is used for setting up buffer swapping.

Table 36. Wake-up time fields

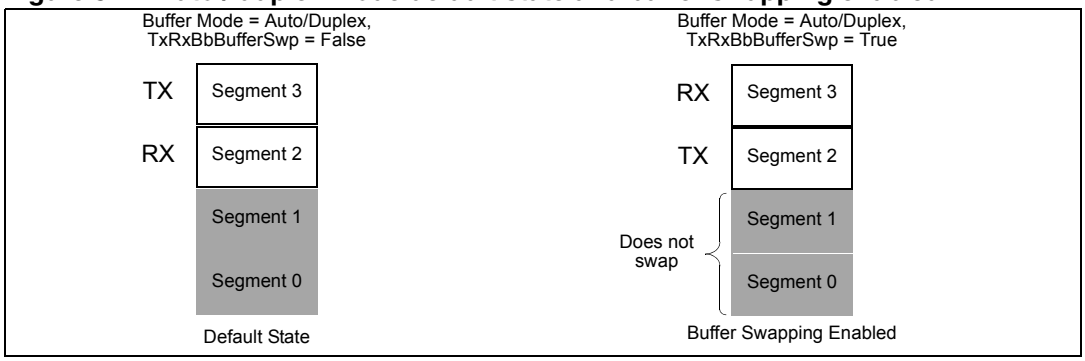
Field	Offset	RW	Description
`SwapBbBuffers`	0x4A	RW brst BbClk	Exchanges RX and TX Baseband Buffers.

- `SwapBbBuffers = 1 (True)` – data for a transmit operation is taken from the receive buffer while the data of a receive operation is stored in the transmit buffer. Enable only for Auto / Duplex mode and Transparent / Duplex mode.
- `SwapBbBuffers = 0 (False)` – data in the buffers belong to their intended direction (default state).

16.2 Buffer swapping in auto / duplex mode

The following shows the default state for Auto / Duplex mode and when buffer swapping has been enabled:

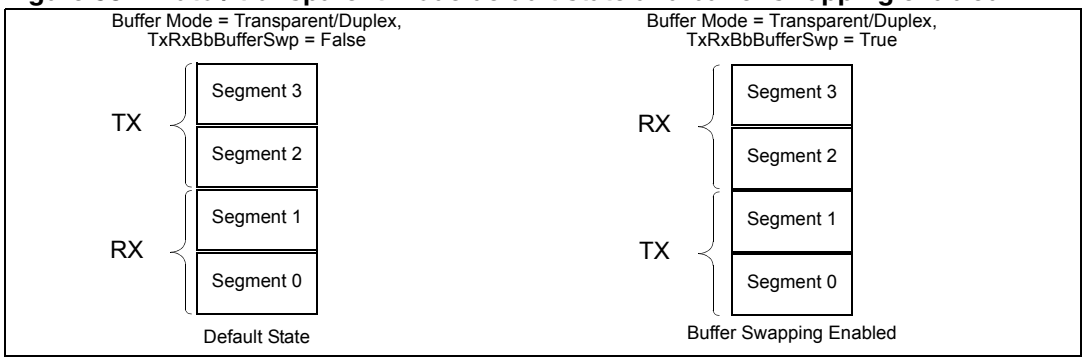
Figure 57. Auto / duplex mode default state and buffer swapping enabled



16.3 Buffer swapping in transparent / duplex mode

The following shows the default state for Transparent / Duplex mode and when buffer swapping has been enabled:

Figure 58. Auto / transparent mode default state and buffer swapping enabled



16.4 Buffer swapping and frame forwarding

To eliminate the overhead of double data transactions over the SPI interface using store and forward algorithm Buffer Swapping can be used. This results in less time wasted with SPI communication and saves electrical energy.

When MACFrames are to be forwarded, the value of `SwapBbBuffers` has to be inverted after a reception but before the next (forwarding) transmission of the MACFrame's (payload) data. This eliminates three operations – reading out the data from the receive buffer, storing it in the external microcontroller, and writing it back in the transmit buffer.

16.5 Buffer swapping and short MACFrames

Buffer swapping can be used if two short MACFrames have to be transmitted or received in a close sequence with a short time gap and the external microcontroller is not able to deliver or consume data in the short time gap between two frames.

The buffer swapping feature eliminates SPI interface transactions for the data of short MACFrames. Short MACFrames in Auto baseband mode are when the payload size is less than or equal to 128 bytes. Short MACFrames in Transparent baseband mode are when the number of transparent bytes is less than or equal to 256 bytes.

When frames have to be transmitted or received in close sequence, the value of `SwapBbBuffers` has to be inverted between the two frames in sequence. Data for both packets can be written to memory before the first one is transmitted or two packets can be received completely before the data of both is read out.

Note: This feature can be used in the Duplex mode and only if the payload size is 128 bytes in Auto baseband mode or if the frame size is 256 bytes in Transparent mode.

17 Buffer control timing

This section describes the control timing for transmit and receive buffers.

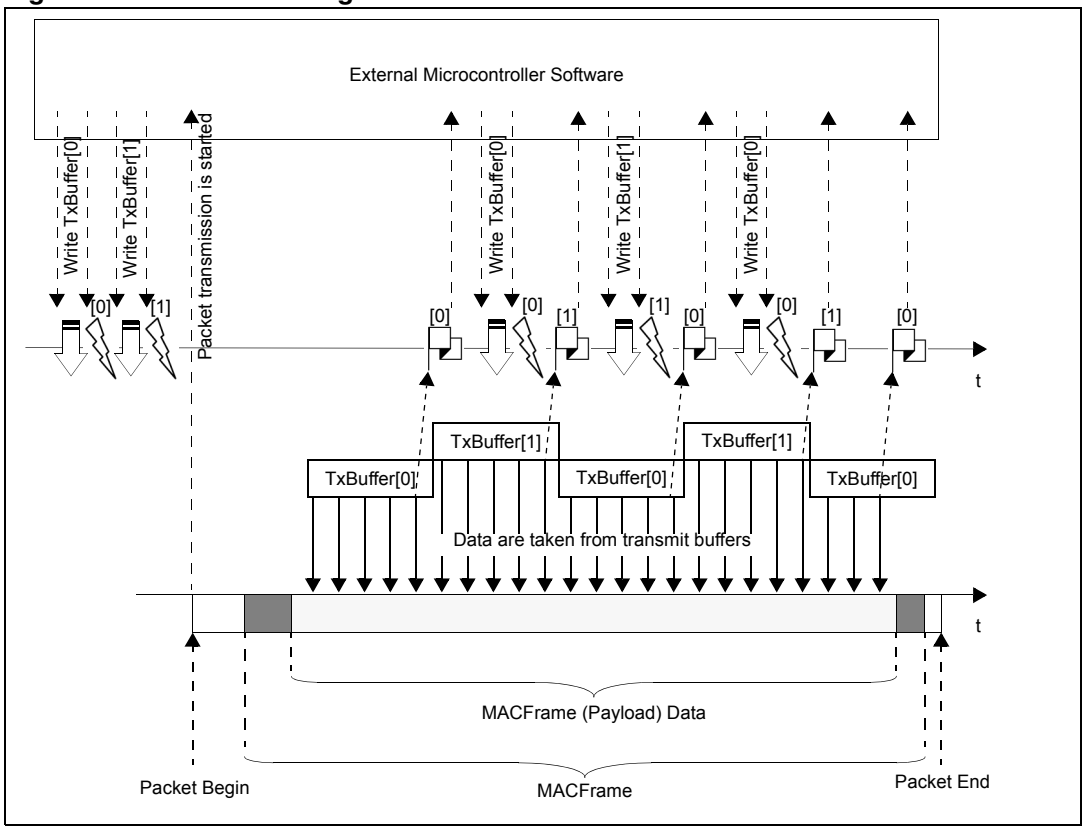
17.1 Transmit buffer control timing

Before transmitting a packet, the external microcontroller software fills up both upper and lower transmit buffer segments with data. After each data transaction, the corresponding buffer command is issued. The software then initiates the transmission of a packet.

Data for transmission is taken from the transmit buffers, but always beginning with the lower buffer segment (`Buffer[0]`). The buffer ready flag indicates that a buffer segment has been emptied and is ready to be filled up again. When the number of MACFrame (payload) data bytes is not an integer number of the buffer segment size, the buffer ready flag is also triggered when the last byte of the frame has been taken from a buffer segment. In this case, it is not required to fill this buffer part up completely, but it is mandatory to issue the buffer command.

A typical transmit buffer control timing is illustrated in *Figure 59*.

Figure 59. Transmit timing



A transmit underrun exception occurs (`TxUnderrun`) when the transmitter tries to send out data from a buffer segment that was not beforehand filled up properly. The transmit operation runs out of data and cannot be completed. The flag informs the software, which usually treats the exception in its protocol stack.

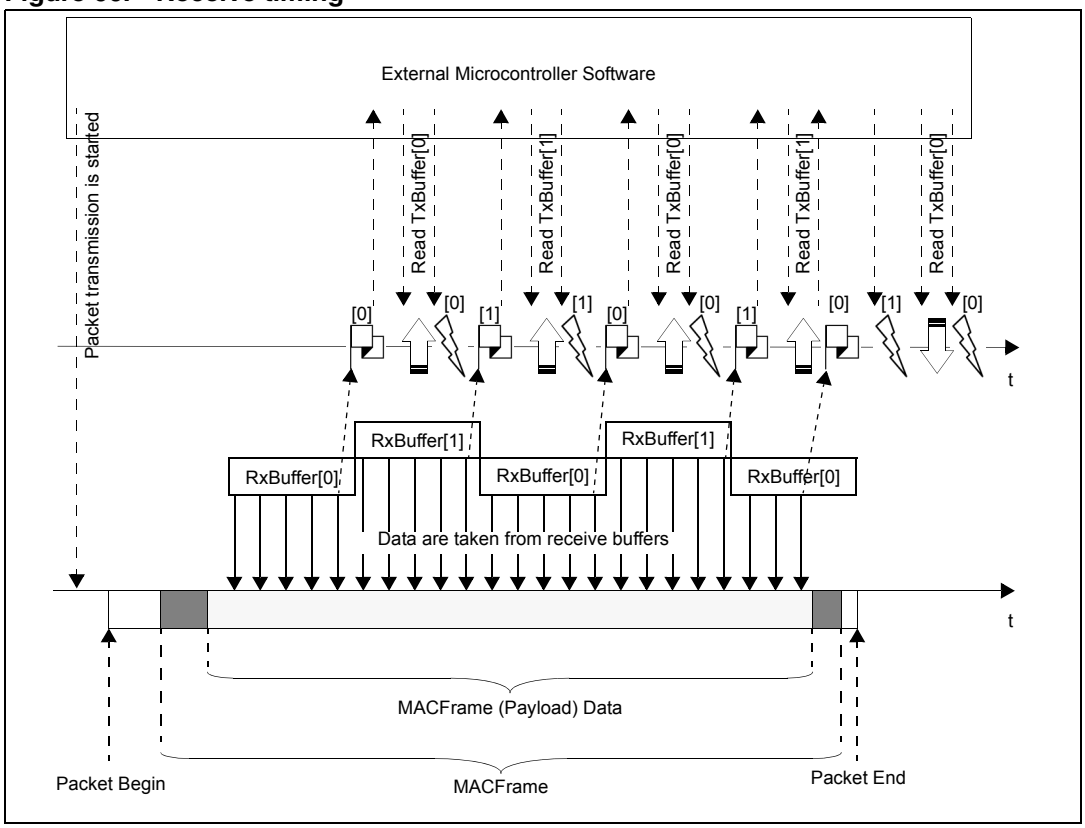
17.2 Receive buffer control timing

Before a packet can be received, software starts the reception of a packet. If the receiver detects a packet, the MACFrame (payload) data is stored in the receive buffer segment, but always beginning with the lower buffer segment (`Buffer[0]`). When the receiver has filled up a buffer segment, the corresponding buffer ready flag is triggered. The corresponding receive buffer can be emptied by software, which was indicated by the flag.

When the last byte has been read from a receive buffer segment, the software issues the receive buffer ready command to indicate to the receiver that the buffer part has been read out completely. If the number of MACFrame (payload) data bytes is not an integer number of the buffer part size, the buffer ready flag is also triggered when the last byte of the frame has been stored in a buffer part. In this case, it is not required to read out this buffer segment completely, but it is mandatory to issue the buffer command.

A typical receive buffer control timing is illustrated in *Figure 60*.

Figure 60. Receive timing



A receive overflow exception occurs (`RxOverflow`) when the receiver tries to store a byte in a buffer part that was not indicated as empty. The receive operation causes a data overflow and cannot be completed properly. The flag informs the software, which usually treats the exception in its protocol stack.

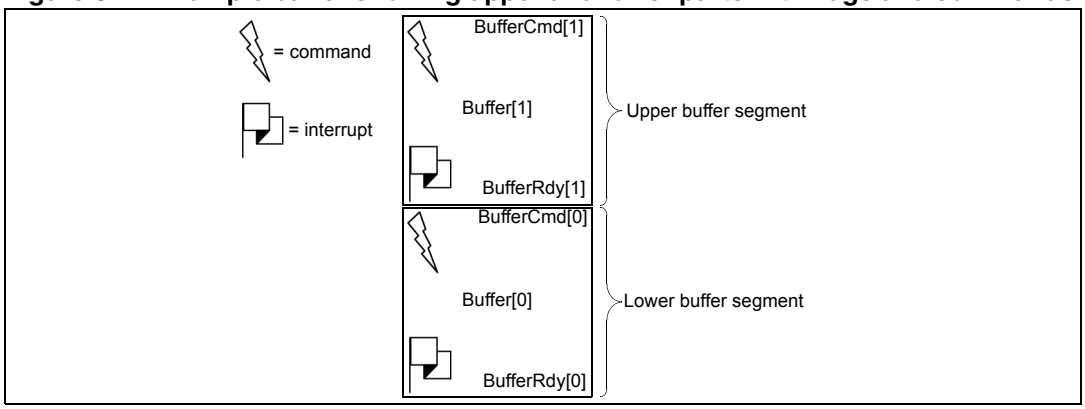
18 Buffer access synchronization

This section describes the flags and commands used for buffer access synchronization.

18.1 Flags and commands

To avoid underrun or overflow exceptions of TX or RX data, the delivery or consumption of data by an external microcontroller must be synchronized to the transmit or receive data rate. Consequently, each TX and RX buffer is divided into two parts or blocks of equal length – an upper part and a lower part. Additionally, each buffer has a flag and a command to control the data traffic to and from the buffers. An example buffer is illustrated in *Figure 61*.

Figure 61. Example buffer showing upper and lower parts with flags and commands



Note: The size of the buffer depends on the selected buffer configuration mode.

A flag-command pair is provided for each buffer segment (lower and upper) and for each direction (transmit and receive).

18.2 Receive and transmit flags

Flags indicate to the software that buffers are ready for a data transfer and are not accessed by the transceiver baseband. Usually flags are enabled as interrupts so that the software can react to a triggered flag as quickly as possible. The following tables lists the receive and transmit flags.

Table 37. Receive flags

Flag	Description
`RxBufferRdy[0]`	Indicates to the software that `RxBuffer[0]` (lower segment) has been filled up with data by a receive operation.
`RxBufferRdy[1]`	Indicates to the software that `RxBuffer[1]` (upper segment) has been filled up with data by a receive operation.
`RxOverflow`	Indicates to the software that a buffer part has been accessed by the receiver, but was not ready to store data (a receive buffer command was not issued in time).

Table 38. Transmit flags

Flag	Description
TxBufferRdy[0]	Indicates to the software that TxBuffer[0] (lower segment) has been emptied by a transmit operation and is ready to store new data.
TxBufferRdy[1]	Indicates to the software that TxBuffer[1] (upper segment) has been emptied by a transmit operation and is ready to store new data.
TxUnderrun	Flag which indicates to the software that a buffer part has been accessed by the transmitter but was not ready for transmission (means that a transmit buffer command was not issued in time).

18.3 Receive and transmit commands

Commands are issued after a data transaction and indicate to the transceiver baseband that data is ready for transmission or buffers are ready to store received data. The following tables lists the receive and transmit commands.

Table 39. Receive commands

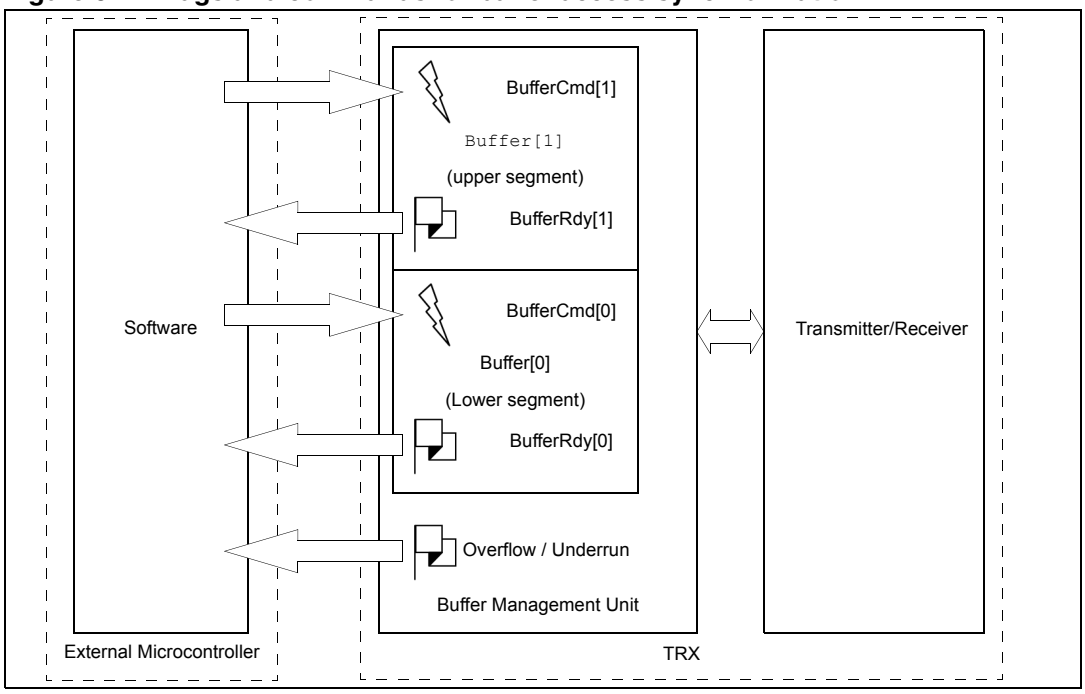
Command	Description
RxBufferCmd[0]	Indicates to the transceiver that RxBuffer[0] (lower segment) has been read out and the buffer is now ready to store new data of a receive operation.
RxBufferCmd[1]	Indicates to the transceiver that RxBuffer[1] (upper segment) has been read out and the buffer is now ready to store new data of a receive operation.

Table 40. Transmit commands

Command	Description
TxBufferCmd[0]	Indicates to the transceiver that TxBuffer[0] (lower segment) has been filled up and data is now ready for transmission.
TxBufferCmd[1]	Indicates to the transceiver that TxBuffer[1] (upper segment) has been filled up and data is now ready for transmission.

The following diagram shows the flags and commands for buffer access synchronization.

Figure 62. Flags and commands for buffer access synchronization



Note: Flags and commands are always related to their buffer segments, independent of buffer size. Also, transmit buffer overflows are severe software errors and are not implemented.

19 Bit processing

This section describes the methods used in the TN100 transceiver for bit error detection, forward error correction, bit scrambling, retransmission, and encryption.

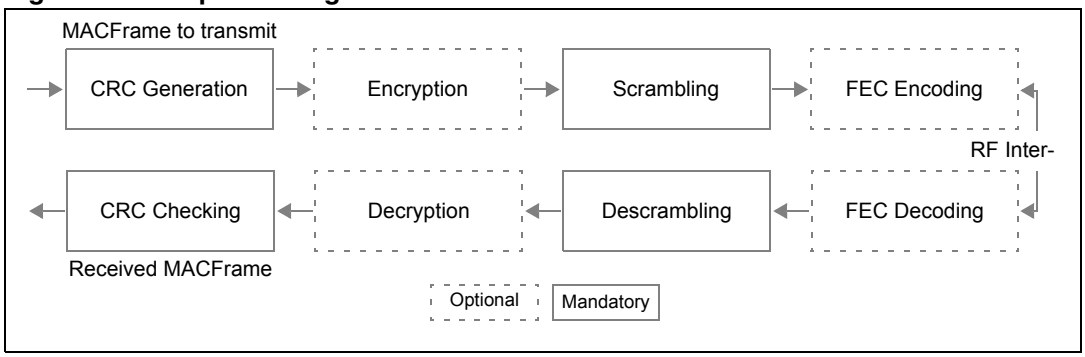
19.1 Bit stream processes

Before application data is sent out over the air interface, several bit manipulations are performed in the transmitter to increase reliability and security. These include:

1. Cyclic Redundancy Check (CRC), which is added to the MACFrame for error detection.
2. Encryption, which is optionally applied to the Data and CRC2 fields of the MACFrame.
3. Bit scrambling with a PRN-sequence.
4. Forward Error Correction (FEC) coding, which is optionally applied.

In the receiver, the reverse process is carried out. The following illustrates the processes:

Figure 63. Bit processing in transmitter and receiver

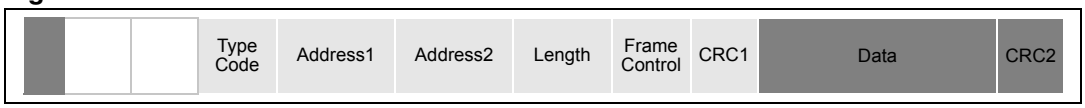


Note: CRC and bit scrambling processes are mandatory, while FEC and encryption can be enabled by the application.

19.2 Cyclic redundancy check (CRC)

The purpose of the CRC scheme is to detect bit errors occurring during transmission. The CRC field is a 16 or 32 bit field containing a Cyclic Redundancy Check. Two CRC fields are defined in the MACFrame: CRC1 and CRC2.

Figure 64. CRC fields in MACFrame



19.2.1 CRC1 field

The CRC1 value is calculated over the following MACFrame header fields: TypeCode, Address1, Address2, Length, and FrameControl. It is not calculated over the ScrInit and Crypt fields. The CRC1 is appended to MACFrame header fields and transmitted with the LSB first.

19.2.2 CRC2 field

The CRC2 value, which is optional, is calculated over the Data field of Data and Brdcast packets. Different polynomials of 16 or 32 bit lengths are defined and can be selected for the CRC2. The CRC2 is appended to the Data field of the MACFrame and transmitted with the LSB first.

19.2.3 CRC types 1, 2, and 3

Three CRC types are defined – CRC type 1, CRC type 2, and CRC type 3. The CRC1 field uses only CRC type 1 whereas the CRC2 field can use any of the three CRC types or use none. A user application can select an appropriate CRC type to satisfy error detection requirements or specific customer demands.

- CRC type 1 (ISO/IEC 3309, CCITT X.25,X.75, ETS 300 125)

Length	16 bit
Polynomial	$2^{16}+2^{12}+2^5+1$
Initial value of LFSR	0xFFFF
CRC	Ones complement of the LFSR
Check value of LFSR	0xF0B8

- CRC type 2 (IEC 60870-5-1)

Length	16 bit
Polynomial	$2^{16}+2^{13}+2^{12}+2^{11}+2^{10}+2^8+2^6+2^5+2^2+1$
Initial value of LFSR	0x0000
CRC	Ones complement of the LFSR
Check value of LFSR	0xA366

- CRC type 3 (CCITT-32, Ethernet)

Length	32 bit
Polynomial	$2^{32}+2^{26}+2^{23}+2^{22}+2^{16}+2^{12}+2^{11}+2^{10}+2^8+2^7+2^5+2^4+2^2+2+1$
Initial value of LFSR	0xFFFFFFFF
CRC	Content of the LFSR
Check value of LFSR	0xCBF43926

19.3 Bit scrambling

Before transmission, all fields of the MACFrame, except the ScramblerInit field, are scrambled with a PRN-sequence to randomize the data into highly redundant patterns and to minimize the probability of large zero patterns. At the receiver, the received data is descrambled using the same PRN-sequence that was generated in the transmitter.

The PRN-sequence is generated with the polynomial g(D)= D7+D4+D0 (221 in octal representation) and is subsequently EXORed with the MACFrame. Before each transmission, the shift register must be initialized with a 7 bit value by the transmitting station. The value should be changed for every new packet, even for retransmissions.

Note: No scrambling is performed when an initialization value of zero is programmed.

The initial value of the shift register is loaded in the ScramblerInit field of the packet (LSB first) and sent over the air interface. During reception of a MACFrame, the shift register of the receiver descrambler is initialized with the bits from the ScramblerInit field. Consequently, the remaining fields are subsequently EXORed with the same PRN-sequence.

19.4 Forward error correction (FEC)

Using Forward Error Correction (FEC) scheme on the MACFrame reduces the number of retransmissions. However, in a reasonable error-free environment, FEC gives unnecessary overhead that reduces the throughput. Therefore, the user application must decide whether or not it should be applied. If FEC is applied, all MACFrame fields are elements of the FEC algorithm.

The FEC scheme is a (7,4) shortened Hamming code and consists of 4 data bits and 3 parity bits. The parity bits are generated with the polynomial g(D)=D3+D+1. The LFSR generating the parity bits is illustrated below.

The code is able to correct single bit errors and detect double bit errors within a codeword. Consequently, a 14 bit code word can contain 2 bit errors even if the bit errors are consecutive. This is achieved through the interleaving scheme.

The receiver informs the LLC about the number of bit errors that have occurred during the reception and correction of a Data packet. This bit error number combined with the Length field is an indication of the BER at the physical interface.

19.5 Encryption

The encryption process is applied to the Data and CRC2 fields of a MACFrame. This process is based on a stream cipher with a symmetric key scheme. The stream cipher sequence bits are XORed with the data bits before transmission or after reception. The cipher stream is re-initialized for every packet (re-)transmission. Since the process is reversible, the same cipher stream is used for encryption and decryption. Consequently, the same input values are required for the generator in both the transmitter and the receiver. The 32 bit clock value and the 48 bit address are publicly known while the 128 bit secret key is generated during the encryption negotiation process and never disclosed.

To keep the probability very low of a successful correlation attack, the same cipher stream is never used twice. The 32 bit clock value is changed (at least incremented by one) for every new packet transmission.

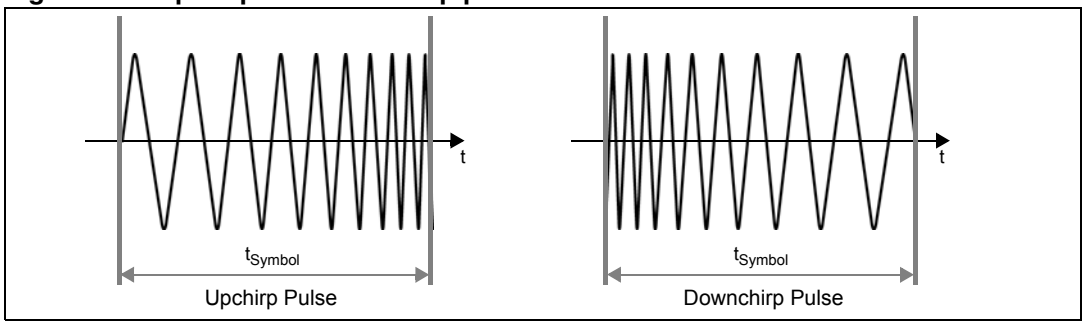
20 Chirp modulation

This section describes describes the TN100 transceiver's modulation technique, Chirp Spread Spectrum (CSS), which was developed by Nanotron.

20.1 Chirp spread spectrum (CSS)

A chirp pulse is a frequency modulated pulse with a frequency that changes monotonic from a lower value to a higher value (called an *Upchirp*) or from a higher value to a lower value (called a *Downchirp*). These two chirp pulses are illustrated in *Figure 65*.

Figure 65. Upchirp and Downchirp pulses



The difference between these two frequencies is approximately the bandwidth *B* of the chirp pulse. For this specification, Upchirps and Downchirps have the following characteristics:

- Chirp duration: t_{Symbol} = 0.5 µs, 1 µs, 2 µs, 4 µs
- Symbol period: $t_{SPeriod}$ = 0.5, 1, 2, 4 µs @ 80 MHz or 2, 4 µs @ 22 MHz
- Frequency bandwidth: B_{Chirp} = 80 MHz, 22 MHz

Note: Chirp duration is the time when a signal is present and the gap is not regarded. For example, chirp duration is 0.5 µs and symbol period is 1 µs. Then symbol duration consists of a 0.5 µs chirp signal and a 0.5 µs gap.

Using these chirp pulses, several different modulation systems can be selected to represent binary symbols. These include:

- Upchirps and Downchirps
- On/Off Keying (Upchirp/Off and Downchirp/Off)

20.2 Secondary modulation systems

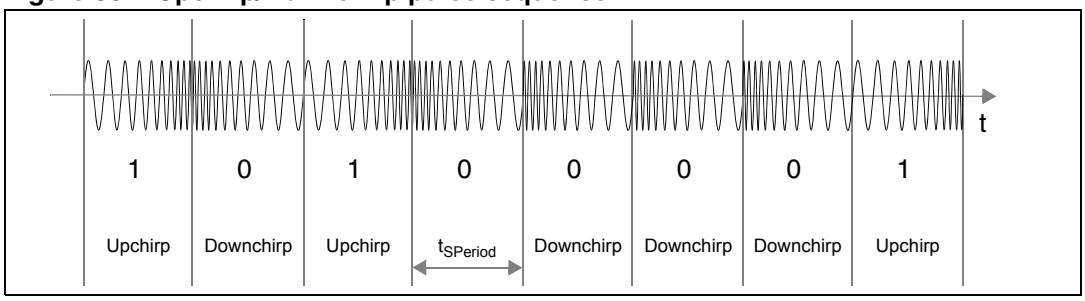
Using Upchirp and Downchirp and On/Off Keying, several secondary modulations systems are possible.

20.2.1 Upchirp/downchirp modulation system

An Upchirp and a Downchirp could be used to represent binary symbols. For example, the Upchirp could represent a bit value of "1" while the Downchirp could represent a bit value of "0".

An example of an Upchirp/Downchirp pulse sequence is illustrated in *Figure 66*.

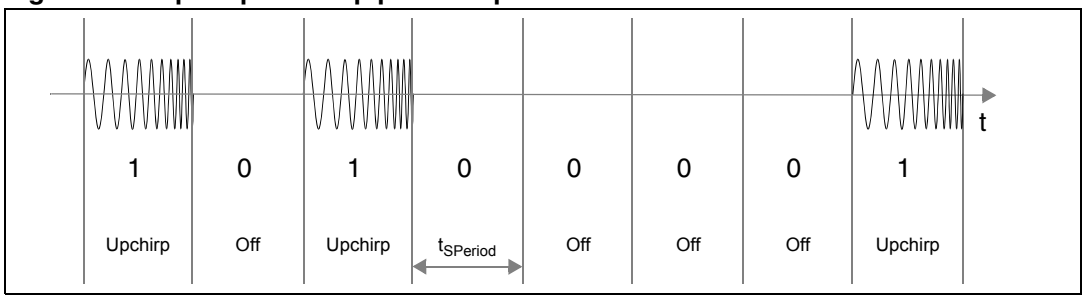
Figure 66. Upchirp/Downchirp pulse sequence



20.2.2 Upchirp/off modulation system

Using On/Off Keying, an Upchirp/Off modulation system can be produced, where an Upchirp pulse could represent a bit value of “1” while the absence of an Upchirp pulse (an “off” chirp pulse) could represents a bit value of “0”. An example of an Upchirp/Off pulse sequence is illustrated in *Figure 67*.

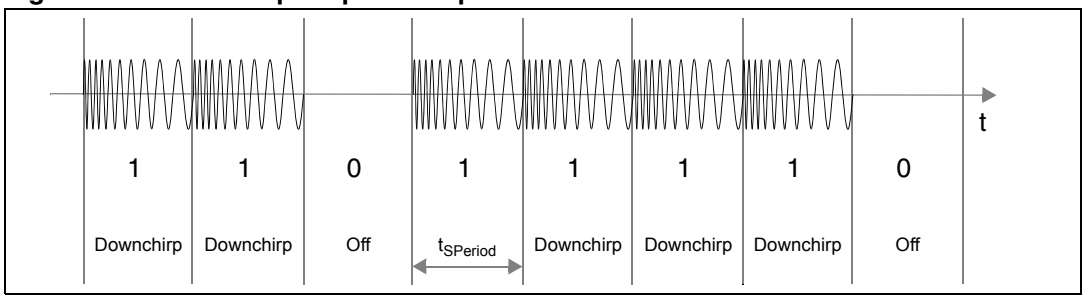
Figure 67. Upchirp/off chirp pulse sequence



20.2.3 Downchirp/off modulation system

Using On/Off Keying, a Downchirp/Off modulation system can be produced, where a Downchirp pulse could represent a bit value of “1”, while the absence of a Downchirp pulse (an “off” chirp pulse) could represents a bit value of “0”. An example of a Downchirp/Off pulse sequence is illustrated in *Figure 68*.

Figure 68. Downchirp/off pulse sequence



20.2.4 Physical channels and the on-off keying modulation systems

The two binary On/Off keying modulation systems (Upchirp/Off and Downchirp/Off) require only Upchirp or Downchirp pulses. Consequently, these two different on-off keying systems can access the same medium with low interference (*0 < cross correlation < 0.5*) to each other.

Both systems can operate in the same frequency channel, in the same physical range, and at the same time but remain independent of each other. They can, therefore, be regarded as separated physical channels.

21 Local oscillator

This section describes how to calibrate Local Oscillator frequency.

21.1 Purpose of the local oscillator

Switched capacitors are used to calibrate the Local Oscillator (LO) frequency. The total capacitance of 22 capacitors plus the capacitance of the oscillating circuit determine the frequency. Each of the twelve capacitors can be separately switched on and off.

The capacitor values are stored for both receive and transmit modes inside the Local Oscillator controller. A single control signal permits the quick tuning of the Local Oscillator to the corresponding target frequency.

During normal operation, the LO controller powers the Local Oscillator and sets the receive or transmit frequency automatically, depending on the operation of the MAC controller.

To compensate for die temperature and supply voltage drifts, as well as for fabrication process variations, the Local Oscillator controller supports an automatic Local Oscillator frequency adjustment with a minimum of required software interaction. The Local Oscillator frequency is compared with the internal crystal quartz frequency while the implemented successive approximation algorithm determines the capacitor switch values (on or off), depending on the comparison result and a value for the target frequency. The capacitor switch values (on or off) can be read out and written to again if the automatic readjustment is not required after a power down cycle.

21.2 Calibrating the local oscillator frequency

The software interaction of the Local Oscillator frequency calibration is as follows:

1. Start up the baseband clock and reset the digital baseband controller, if required.

Note: Steps 2 and 3 can be performed simultaneously.

2. Enable the LO circuitry, the LO divider, and the LO adjustment clock as follows:
   a) Set `EnableLO = True`.
   b) Set `EnableLOdiv10 = True`.
3. Switch the Local Oscillator into transmit mode by setting `UseLoRxCaps = False`.
4. Wait 24 µs.
5. Set the 16 bit value for the TX target frequency `fTarget` as follows:

$$\text{LoTargetValue} = \texttt{round}\left(\langle\frac{\text{LoDiv10}}{16} - 12\rangle \cdot 8192\right)$$

   The adjustment procedure starts automatically when writing to the upper bits of the value.

   Frequency example: `fTarget` = 2.44175 GHz:

Table 41. Value for fTarget

fTarget	RfLoAdjIncValue
2.44175 GHz	0x685A

6. Wait 13 ms (baseband timer could be used).

Note: When fast tuning mode is used, then wait time is reduced to 6 ms. See Section 26.2.24: RF local oscillator controls on page 148 for details on using fast tuning mode.

7. Optionally, the resulting value LoTxCapsValue can be put into a variable to be used after power down mode. During powering up, this value can then be reused:
8. Disable the LO adjustment clock, LO divider, and the LO circuitry as follows:
   a) Set `EnableLOdiv10 = False`.
   b) Set `EnableLO = False`.
9. Repeat the procedure for tuning the receiver. Use LoRxCaps = True for tuning RX.

21.2.1 Fields for updating the local oscillator

The following fields are used for updating the Local Oscillator.

Table 42. RTC and TimeB packets

Field	Offset	R/W	Description
`EnableLO`	0x42	WO brst BbClk	Enables the Local Oscillator. Required only for tuning the Local Oscillator and for programming the Chirp Sequencer (CSQ) RAM.
`EnableLOdiv10`	0x42	WO brst BbClk	When enabled, the LO frequency divided by 10 is delivered to the digital part. Required only for tuning the Local Oscillator and for programming the Chirp Sequencer (CSQ) RAM.
`UseLoRxCaps`	0x1C	WO brst BbClk	Switch between TX and RX Caps for LO caps tuning. This field must be set before starting the LO caps tuning.
`LoTargetValue`	0x1D – 0x1E	WO brst BbClk	16 bit target value for the Local Oscillator frequency adjustment. This value is used to determines the target LO center frequency. The LOTargetValue is determined as follows: $\lceil (\{LOdiv10/(16MHz)\} - 12) \cdot 8192 \rceil$ Before setting this value, the following must be enabled: Local Oscillator – EnableLO Clock divider – EnableLOdiv10 Also, the appropriate mode (RX, TX) must be selected using the UseLoRxCaps: UseLoRxCaps = 1 (RX mode) UseLoRxCaps = 0 (Tx mode) A write operation to the upper byte starts the tuning algorithm automatically.
`LoTxCapsValue`	0x19 – 0x1B	RW brst BbClk	Reads or writes the 22 capacitors of the Local Oscillator for transmit frequency. The correct capacitor values can be read after the Local Oscillator has been tuned. Default Value = 0x200040

22 RF transmitter

This section describes the adjustment of the Baseband filter frequency, the power amplifier bias current, and the output power of the transmitter.

22.1 Adjusting the baseband filter frequency

Why Adjust this Frequency?

The frequency of the Baseband filter (chirp DAC filter) must be adjusted because of process deviations of the resistor values and capacitor values.

Realization

- A fixed relationship exists between the RC-Oscillator (RCOSC) frequency and the Baseband filter frequency because both circuits use the same resistor and capacitor types.
- The user can set the 4 bit value `ChirpFilterCaps` to adjust the RC-oscillator/Baseband filter frequency.
- `ChirpFilterCaps` is used for adjusting the RCOSC frequency in comparison with the system clock.
- The algorithm has to be implemented in software.

To Adjust the RC-Oscillator Baseband Filter Frequency[(b)]

1. Set the start value to `ChirpFilterCaps = 0b0110`.
2. Enable the RC-Oscillator (RCOSC) by setting `TxFctEn = True`.
3. Ignore at least 4 cycles of RCOSC output frequency by waiting 20 µs (baseband timer can be used).
4. Count the cycles of the baseband system clock for next c_{ref} cycles of RCOSC as follows:
   a) Start the counter by setting `StartFctMeasure = True`.
   b) Wait at least .
   $$\frac{1}{f_{ref}}$$
   c) Read out the counter value from the `FctPeriod` field.
   d) Run this sequence c_{ref} times, accumulating the counter value after each run.
5. Decision (for c_{ref} RCOSC cycles):
   a) If the number of system cycles are greater than the following:
   $n1 = c_{ref} \cdot 43$ (For $c_{ref} = 4$, $n1 = 172$)
   then decrease value for `RfTxChirpFilterCaps`.

b. Cycle reference frequency definitions:

Variable	*Definition*	*Value*
c_{ref}	Cycle reference: number of RCOSC reference cycles.	≥ 4
f_{ref}	Frequency reference: reference frequency of RCOSC.	400 kHz

b) If the number of system cycles are less than the following:

$n2 = c_{ref} \cdot 39$ (For $c_{ref} = 4$, $n2 = 156$)

then increase value for `RfTxChirpFilterCaps (ChirpFilterCaps)`.

c) If the counter value from step 4d above is between n1 and n2, then stop the calibration.

or if $n \leq 172$ after the previous counter value was > 172, or if $n \geq 156$ after the previous counter value was < 156, then:

Set `FctClockEn = False`.

22.2 Setting the power amplifier bias current

To set the PA Bias Current

- Set the following:

Table 43. Setting PA bias current

Switch	Setting
RfTxPaBiasAdj `(TxPaBias)`	000

22.3 Adjusting the transmitter output power

22.3.1 Output power control

The typical characteristics of the output power control (as set by RfTxOutputPower) measured at the SMA connector of a typical RF Module is shown in *Figure 69*. The typical output power values are listed in *Table 44 on page 96*.

Figure 69. Output power control measured at SMA connector of RF Test Module

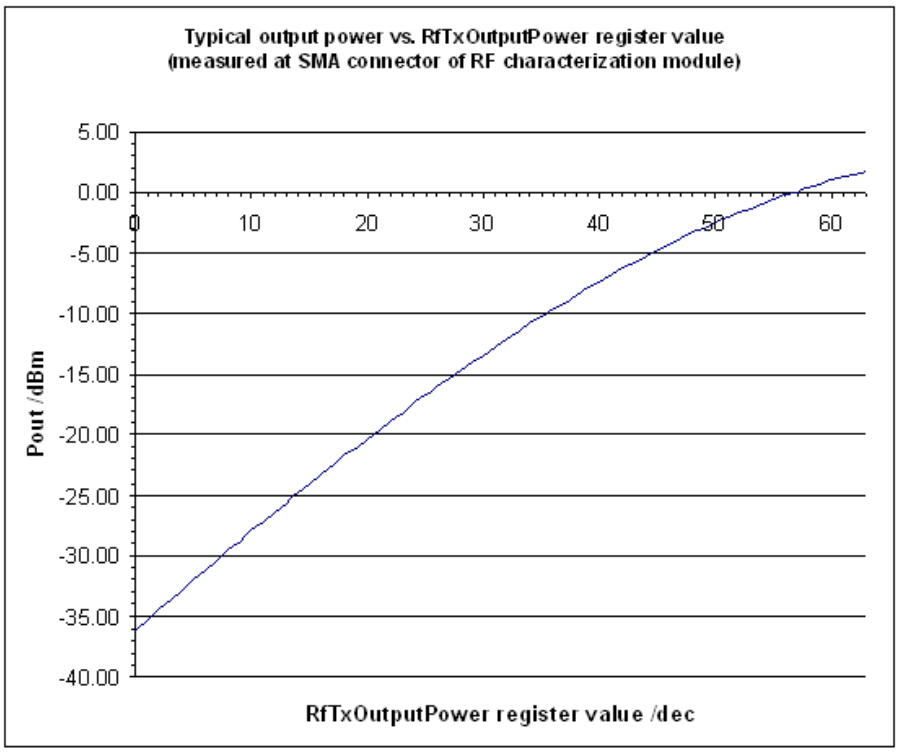


Table 44. Typical output power for RfTxOutputPower register values

Register Value	Pout / dBm	Register Value	Pout / dBm	Register Value	Pout / dBm	Register Value	Pout / dBm
0	-36.20	16	-23.28	32	-12.07	48	-3.33
1	-35.30	17	-22.48	33	-11.41	49	-2.87
2	-34.47	18	-21.75	34	-10.80	50	-2.45
3	-33.65	19	-21.03	35	-10.20	51	-2.05
4	-32.83	20	-20.31	36	-9.61	52	-1.66
5	-32.02	21	-19.60	37	-9.03	53	-1.28
6	-31.21	22	-18.89	38	-8.46	54	-0.92
7	-30.41	23	-18.19	39	-7.91	55	-0.57
8	-29.54	24	-17.44	40	-7.31	56	-0.23
9	-28.70	25	-16.70	41	-6.74	57	0.12
10	-27.92	26	-16.02	42	-6.22	58	0.42
11	-27.14	27	-15.36	43	-5.71	59	0.72
12	-26.37	28	-14.70	44	-5.21	60	1.00
13	-25.60	29	-14.04	45	-4.73	61	1.28
14	-24.85	30	-13.40	46	-4.26	62	1.54
15	-24.09	31	-12.76	47	-3.80	63	1.79

22.3.2 Fields for adjusting the RF Transmitter

The following fields are used for adjusting the RF Transmitter.

Table 45. RTC and TimeB packets

Field	Offset	R/W	Description
ChirpFilterCaps	0x27	WO brst BbClk	Switches the capacitors of the chirp filter to adjust the FCT clock to 400 kHz.
EnableTx	0x27	WO brst BbClk	Enables the transmitter for tuning. It manually starts the transmitter.
FctClockEn	0x27	WO brst BbClk	Switches on the FCT generator. This is required for the chirp filter calibration.
StartFctMeasure	0x27	WO brst BbClk	Starts the tuning algorithm. It initiates an FCT count cycle. FctClockEn must be set to 1 before starting the measurement cycle.
LoTxCapsValue	0x19 to 0x1B	RW brst BbClk	Reads or writes the 22 capacitors of the Local Oscillator for transmit frequency. The correct capacitor values can be read after the Local Oscillator has been tuned. Default Value - 0x200040
TxPaBias	0x43	WO brst BbClk	Adjusts the PA bias current to compensate process deviations.
TxOutputPower0	0x44	WO brst BbClk	Sets the transmitter output power for data (Data) packets, time beacon (TimeB) packets, and broadcast (BrdCast) packets. Default Value = 0x3F

23 Media access control methods

This section describes various media access methods, including Direct Access (DA), Random Access (CSMA/CA) and TDMA.

23.1 Symbol definitions

The following symbol is used in this section:

- MacArqCntMax

Note: These symbols are defined in Appendix A: Attributes and constants on page 221.

23.2 Direct access (DA) mode

A very simple way to access the media is with the Direct Access (DA) mode. Using this method, a station that wants to access the media does not sense the carrier nor wait a specific period of time before accessing the media. However, collisions may occur with transmissions from other stations or with noise sources using the same frequency bandwidth and located in the same range. Direct access is, therefore, intended for protocols where the probability of collisions is very low. Applications of Direct Access include master-slave protocols and TDMA:

- Master-Slave Protocols - A single master station allocates bandwidth for a slave station and it assumes that there are no competitors for the media other than this master. The master then polls a slave to request a Data packet. Only that polled station can transmit a Data packet. The master station then waits for the Data packet a time-out interval and then polls this slave again or else polls other slave stations. The poll command is transmitted to the slave via an LCMP command in a Data packet.
- TDMA - Time Slotted Media Access achieves bandwidth reservation through the use of dedicated time slots.

Note: For more details, see Section 23.4: Time slotted access (TDMA) on page 106.

23.3 Random access (CSMA/CA) mode

23.3.1 General description

Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) is an access method that promotes fairness as all stations have equal probability of gaining access to the media. Fairness is maintained because each station must recontend for the media after every transmission.

CSMA works as follows:

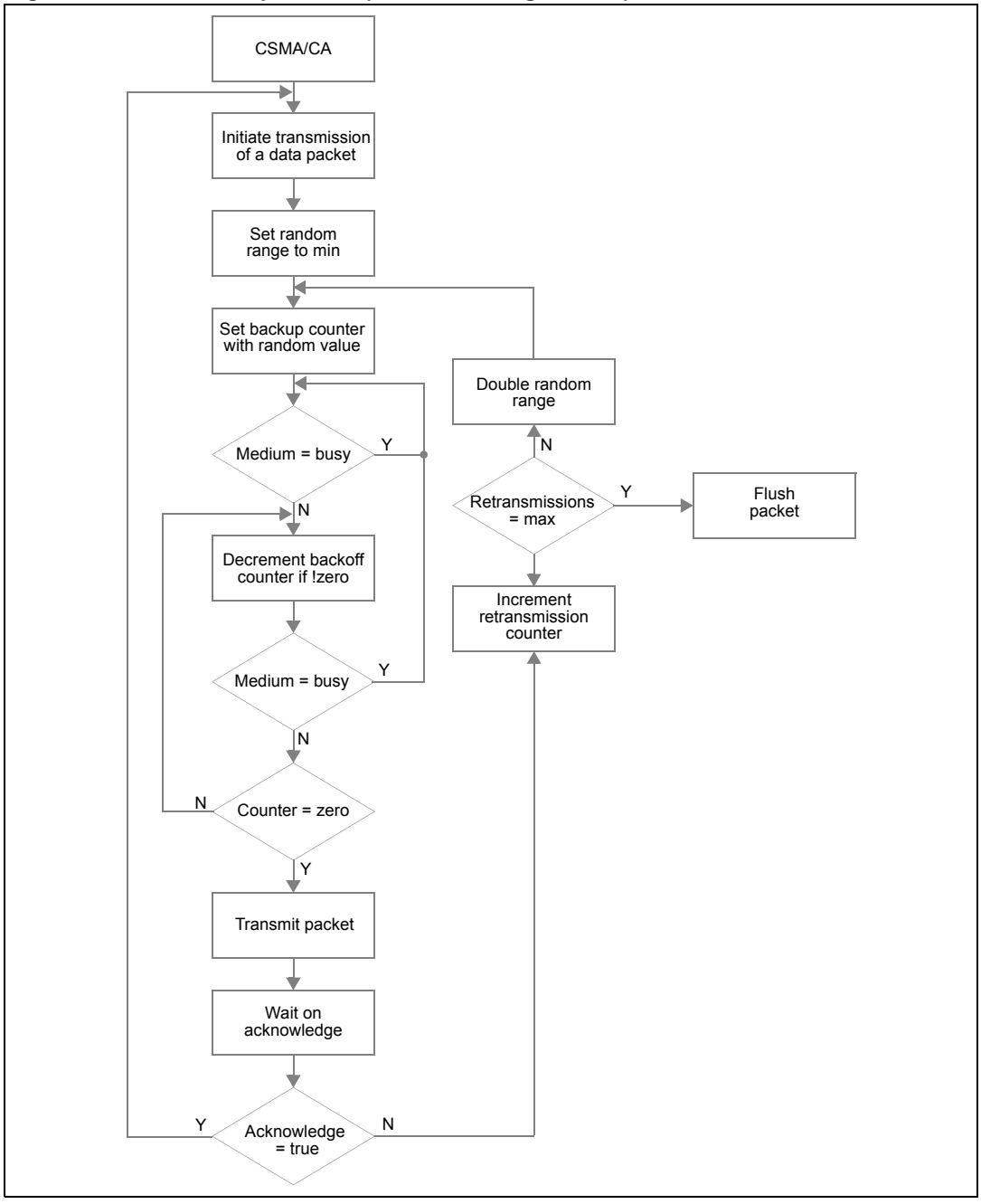
1. A station that has data to transmit first senses the channel (CSMA) for a certain duration to determine if the media is busy before transmitting a message.
2. If the channel is busy, the station attempts a retransmission at a later time.
3. If the channel is idle, the station waits a random period of time (a “backoff” time) and then attempts again to acquire the media. This random backoff scheme avoids collisions (CA) as much as possible.

4. The acquisition of the media is followed by the transmission of the Data packet.
5. This transmission is then acknowledged by the destination station with an Ack packet to allow the transmitting station to determine if the transmission was successful. If an Ack packet was not received, the transmission was deemed not successful.
6. This procedure is repeated until the transmission has been successful, a retransmission counter is exceeded, or a time-out is reached.

Note: For more details, see Section 23.3.4: Retransmissions on page 101.

The scheme for CSMA/CA is shown in *Figure 70*.

Figure 70. CSMA/CA protocol (at transmitting station)



23.3.2 Carrier sensing

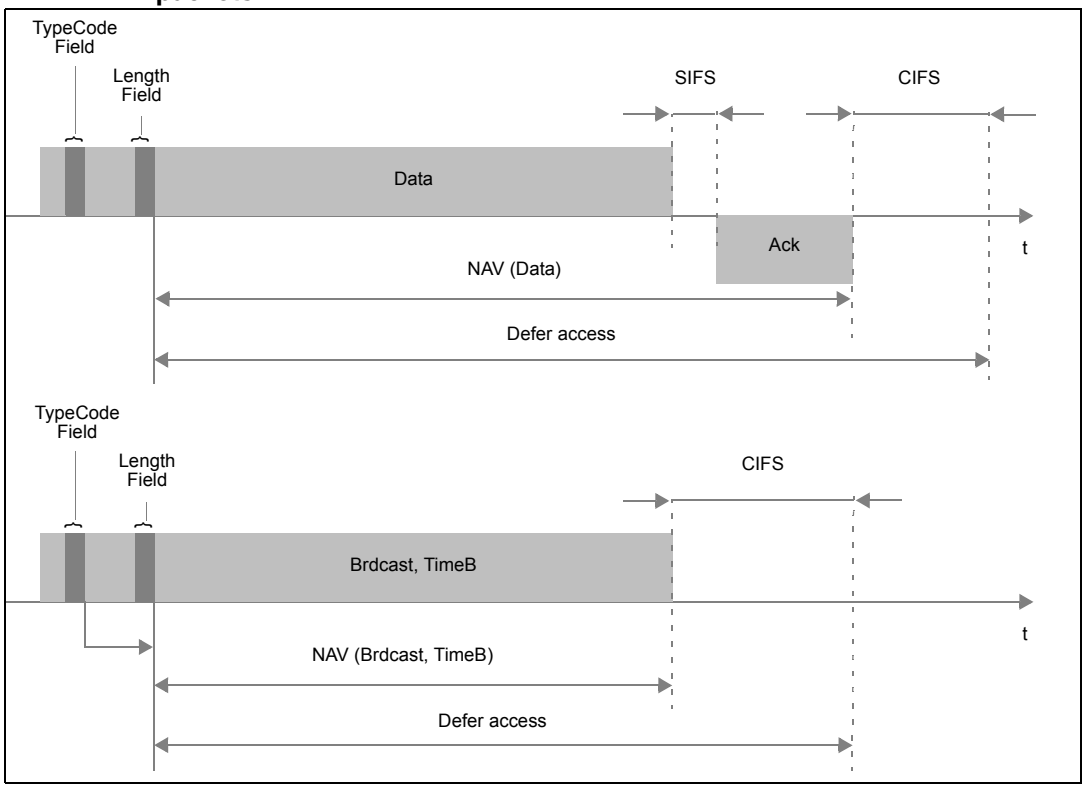
The state of the media is determined by using physical and virtual carrier-sense functions:

- Physical carrier sensing detects activity in the media via a bit detector unit and a RSSI (Receive Signal Strength Indicator). For instance, if the bit detector senses chirp signals, then there is communication activity in the media. The physical carrier sensing unit selects the output of either the bit detector or the RSSI or both. A station will continue sensing the media until the media has been idle for at least a CIFS (Carrier sense InterFrame Space) period.
- Virtual carrier sensing predicts if the media is allocated by detecting the TypeCode and Length fields in packets transmitted by any station. These fields are used to indicate the amount of time required to complete a transmission and are used by stations to adjust their NAVs (Network Allocation Vector). A NAV is an indicator of the amount of time that must elapse until the current transmission is complete and when the media can be sampled again to determine if the media is idle.

  A station updates its NAV with the indication from the Length field. If the error detection over the MACFrame fails (that is CRC1 fails), then the NAV is set to “0” and physical carrier sensing is applied until the NAV can be updated again.

If either the physical or virtual carrier sensing mechanisms indicate that the channel is busy, then the channel is marked busy. Carrier sensing for Data, Brdcast, and TimeB packets is illustrated in *Figure 71*.

Figure 71. CSMA/CA carrier sense timing diagram of Data, Brdcast and TimeB packets

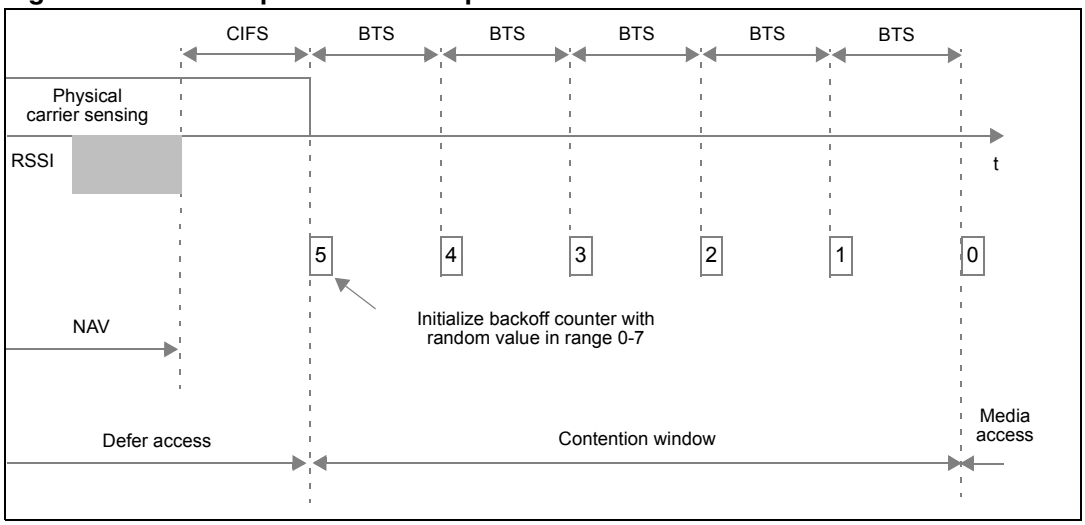


The NAV can also be used as a mechanism for power saving. A station can be powered down for a NAV period of time. However, if a station is powered down, it is unable to receive packets and, consequently, cannot update its NAV.

23.3.3 Backoff time

If the media is detected idle, the transmitting station waits a specific period of time before it reattempts a transmission. This period is the contention window (CW) and is determined by a backoff counter which counts defined time slots, each of which is a Backoff Time Slot (BTS). This is illustrated in *Figure 72*, where BTS = 24 µs.

Figure 72. Backoff procedure example



While the media is idle, a station decrements its backoff counter using physical carrier sensing until the media again becomes busy. While the media is busy, the backoff counter is frozen until virtual and physical carrier sensing detects that the media has again become idle. The state of the media can, therefore, be determined at any time during a BTS.

A station can access the media and transmits a packet only when the backoff counter reaches “0” and if the media is idle.

When a station makes the first transmission attempt, the backoff counter is initialized with a pseudo-random integer in the range of 0 to 7. For every new transmission attempt this range is doubled, which is an exponential backoff. For the second transmission attempt, the backoff counter is initialized with a pseudorandom integer in the range of 0 to 15; for the third attempt, it is initialized in range of 0 to 31; and so on until the range reaches 0 to 255. The pseudo-
random integer is then drawn from the range of 0 to 255 for every new attempted retransmission.

23.3.4 Retransmissions

A retransmission is the repeated transmission of the same Data packet. If a transmitting station does not receive an acknowledgement from a destination station, the Data packet must be retransmitted. This failed acknowledgement can be either the absence of an Ack packet in response to a Data packet or, in the case of a three-way handshake, the absence of a Clr2S packet in response to a Req2S packet. Both cases are regarded as unsuccessful transmission attempts.

Note: See also Section 24.2: Error correction (ARQ scheme) on page 110 and Section 23.3.5: Three-way handshake on page 102.

A transmitting station cannot determine the reason for any unsuccessful transmission attempt. It is unable to detect, for instance, if a collision occurred or if the transmission path was disrupted. In every case, a retransmission is required.

To limit the number of unsuccessful retransmission attempts, a retransmit counter counts the unacknowledged transmissions. When the maximum number of attempts is reached, as defined by the MacArqCntMax attribute, no more retransmissions are made and the LLC is informed that the maximum number of unsuccessful transmission attempts have been reached.

Note: Retransmission is applied only to Data packets. Brdcast and TimeB packets cannot be retransmitted because they do not require an acknowledgement. In the latter case, resent packets are never regarded as retransmitted, even if consecutively sent packets are identical.

23.3.5 Three-way handshake

A three-way handshake is an optional strategy that can be only used with the transmission of Data packets and cannot be used with Brdcast and TimeB packets. It is used to eliminate the so-called hidden node problem and to reduce the amount of wasted bandwidth due to collisions of Data packets.

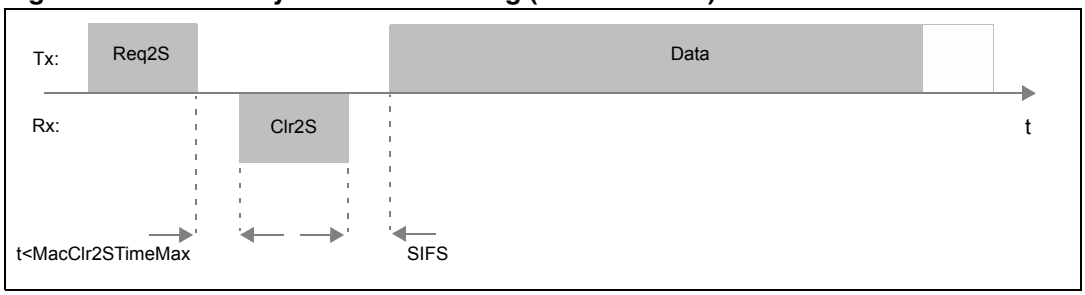
Note: The Req2S/Clr2S mechanism must not be used for Brdcast and TimeB packets because the Req2S has multiple destinations and thus potentially multiple concurrent transmitters of the Clr2S in response.

A three-way handshake works as follows:

1. A station that has data to send transmits a short Req2S packet.
2. All stations in range of the transmitting station hear the Req2S, set their NAVs and defer their transmissions.
3. The destination station waits a SIFS interval (8 µs) and responds with a Clr2S (clear to send packet).
4. All stations in range of the destination station hear the Clr2S, set their NAVs and defer their transmissions.
5. On receiving the Clr2S, the transmitting station assumes that the channel is acquired, waits the SIFS interval, and then transmits the Data packet.
6. The station transmitting the Req2S packet waits for the Clr2S packet an interval defined by MacClr2STimeMax. If during this time it receives and recognizes any other valid packet, such as a Data, Brdcast, TimeB, or Req2S packet, it considers the transmission attempt as unsuccessful. Nevertheless, the recognized packets are processed. If during this time it receives and recognizes an invalid packet, such as an Ack packet, it also considers the transmission attempt as unsuccessful, but in this case the invalid packet is ignored.
7. The station transmitting the Clr2S packet waits for the Data packet until it arrives. If during this time it receives any valid packets, such as Data, Brdcast, TimeB, or Req2S packets, it processes them. If during this time it receives any invalid packets, such as Ack or Clr2S packets, it ignores them.

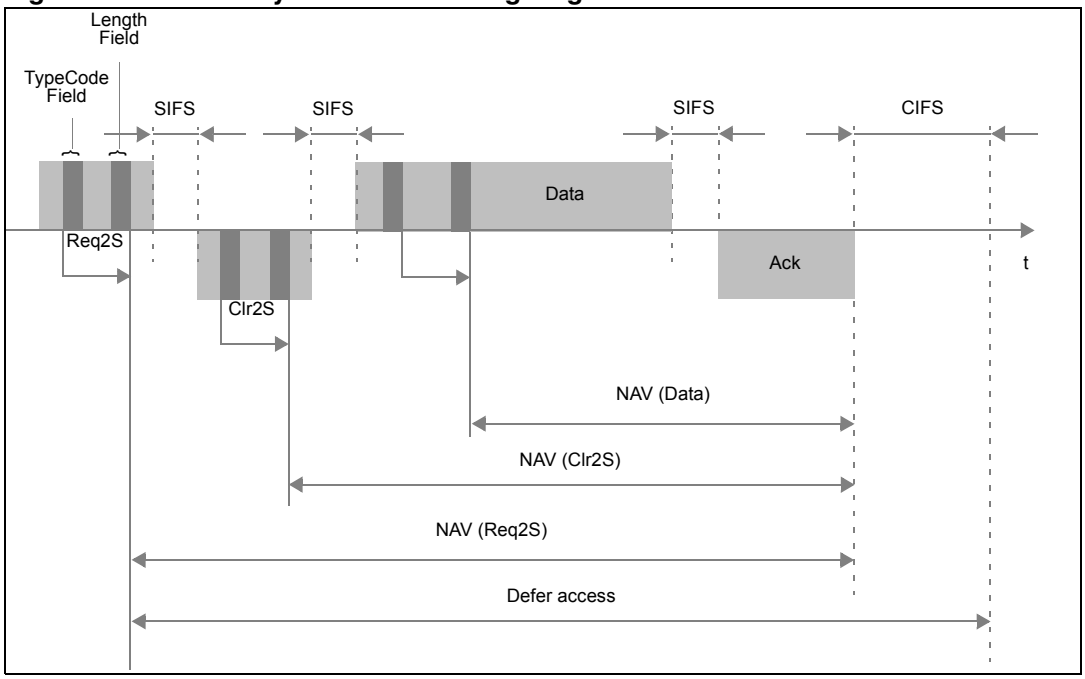
A three-way handshake is illustrated in *Figure 73*.

Figure 73. Three-way handshake timing (at transmitter)



The NAV in combination with a three-way handshake is intended to eliminate the so-called hidden node problem. Carrier sensing in a three-way handshake is illustrated in *Figure 74*.

Figure 74. Three-way handshake timing diagram



Note: The three-way handshake mechanism adds overhead inefficiency because of the additional Req2S and Clr2S packets required for this strategy. Also, if this strategy is used, every transmission of a Data packet will require the use of this mechanism, whether the packets are large or even a fragment of a longer message.

23.3.6 Three-way handshake and ARQ mode

The three-way handshake performs a type of fast collision inference and a transmission path check. Depending on the length of the Data packet, it may be more efficient to use either three-way handshake or ARQ mode.

Three-way handshake requires at total of 272 µs to send the Req2S packet, wait the MacClr2STimeMax, receive the Clr2S packet, and wait the SIFS interval before sending the Data packet to the receiver, as shown in *Table 46*.

Table 46. Three-way handshake timing

Packet/interval	Time
Req2s (144 bits)	144 µs
Clr2S (96 bits)	96 µs
MacClr2STimeMax	24 µs
SIFS	8 µs
Data packet	X µs
Total	272 µs + X µs

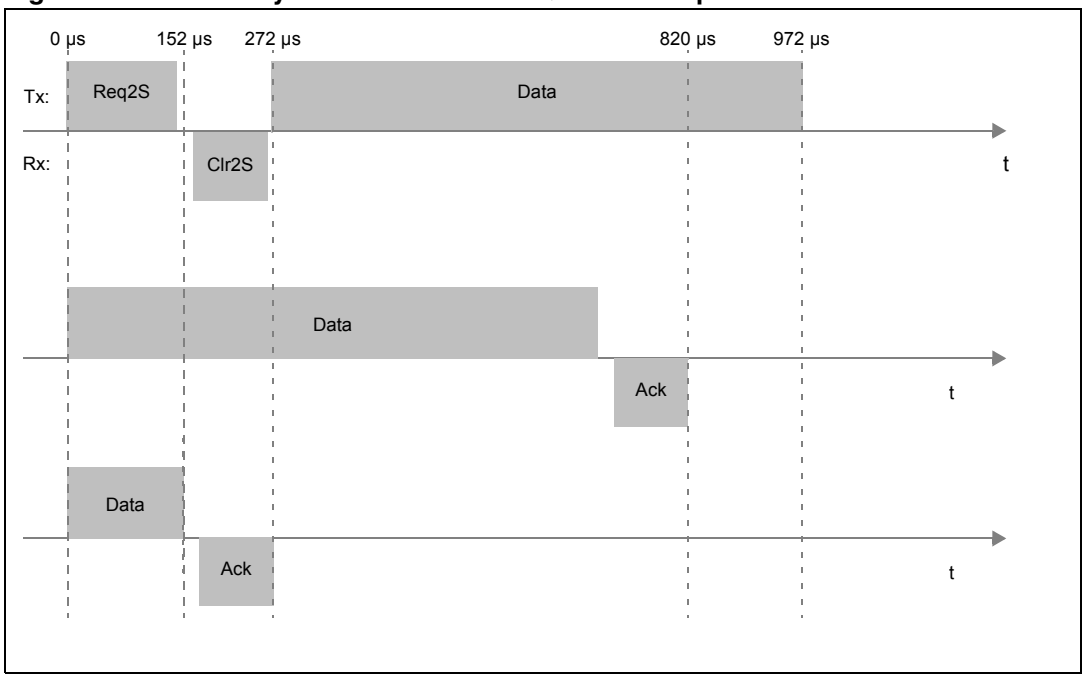
For short Data packets with a length of less than 152 bits (2ary mode), it may be better to use ARQ mode than the three-way handshake. In ARQ mode, collision interference can be detected in a time period of less than 272 µs with a packet of this size. The originating station can therefore repeat the process more quickly (after observing the other media-use rules) if a collision occurred.

For long Data packets with a length of greater than 152 bits, using a three-way handshake rather than ARQ mode will result in collision interference being detected in a shorter time period than with ARQ mode. The process can be repeated more quickly by the originating station (after observing the other media-use rules) if a collision occurred.

Table 47. ARQ mode timing

Packet/interval	Time
MacAckTimeMax	24 µs
Ack packet	96 µs
Data packet	X µs
Total	120 µs + X µs

Figure 75. Three-way handshake and ARQ mode comparison



23.3.7 Adjusting transmit power levels for three-way handshakes

The transmit power level at the air interface is separately adjustable for three-way handshake packets and for other packet types. Reserving bandwidth with Req2S and Clr2S is done using a power level higher than that used for transmitting Data and Ack packets. A higher power level for the former allows remote stations to update their NAVs and reduce the hidden node problem. The lower power level for the latter (optimized for the connection) allows stations to save energy and battery lifetime.

A station configured not to initiate the Req2S/Clr2S mechanism still updates its virtual carrier-sense mechanism with the Length information contained in a received Req2S or Clr2S packet, and will always respond to a Req2S addressed to it with a Clr2S.

23.3.8 Prioritized access

Brdcast and TimeB packets have a higher priority than Data or Req2S packets since their backoff counters are initialized with a random value only in the range of 0 to 7, even for the retransmission of identical packets. The excessive use of these packets can, therefore, decrease the overall performance of a network.

Stations can give their transmission attempts a higher priority by disabling the backoff counter, even for retransmission attempts. However, in this case, the prioritized access does not promote fairness among stations because they would not have an equal probability of gaining access to the media. Consequently, the excessive use of prioritized packets can lead to unpredictable network performance and possibly even a network deadlock.

Nevertheless, the remaining fragments of a message transmission should get a higher priority for the first transmission attempt and for retransmissions. This is achieved by masking out the MSB of the random value before initializing the backoff counter. In this case, this MSB is set to the value “1” for transmitting non-fragmented messages and for the first fragment of a fragmented message, while this MSB is set to the value “0” for

transmitting the remaining fragments of a fragmented message. A fragmented message therefore receives a higher priority than a non-fragmented message without violating the fairness principle.

A station is configurable to switch on and off the fragmentation prioritization scheme.

23.4 Time slotted access (TDMA)

23.4.1 General description

The time scale for transmissions can be divided into multiple time slots (TS). Each time slot provides time for accessing the media and for transmitting at least one packet, including the handshake sequence. Multiple time slots of different durations, repetition periods, and repetition counts can be defined in the time slotted media. The time slotted access scheme has been kept flexible so that adjustments can be made according to performance, application, and specific customer requirements.

To administrate, allocate, and dedicate time slots, a master station is defined which provides time slot information to slave stations upon request. These stations can also request a time slot which the master can grant or deny.

In a network, both the master and the slaves can be predefined. However, if the master is not predefined in a network, stations can negotiate amongst themselves to define one as the master while leaving the remaining stations to become slaves. Therefore, it is possible to predefine all stations (a master and its slaves) or to predefine only the master station. In the latter case, the non predefined stations must scan the network to search for a master.

Note: Time slot administration/negotiation/maintenance procedures are defined in Section 12.2: Link control management on page 66.

23.4.2 Time slots

The basic unit of measurement for times slots is one Time Slot Unit (TSU), which is $\frac{1}{1024}$ s (or 976.5625 µs). A time slot can consist of from 1 to 256 TSUs (up to 250 ms). One TSU is based on 32 RTC (Real-Time Clock) oscillator periods of 32,768 kHz each.

Note: For more details, see Section 23.4.4: Real-time clock on page 108.

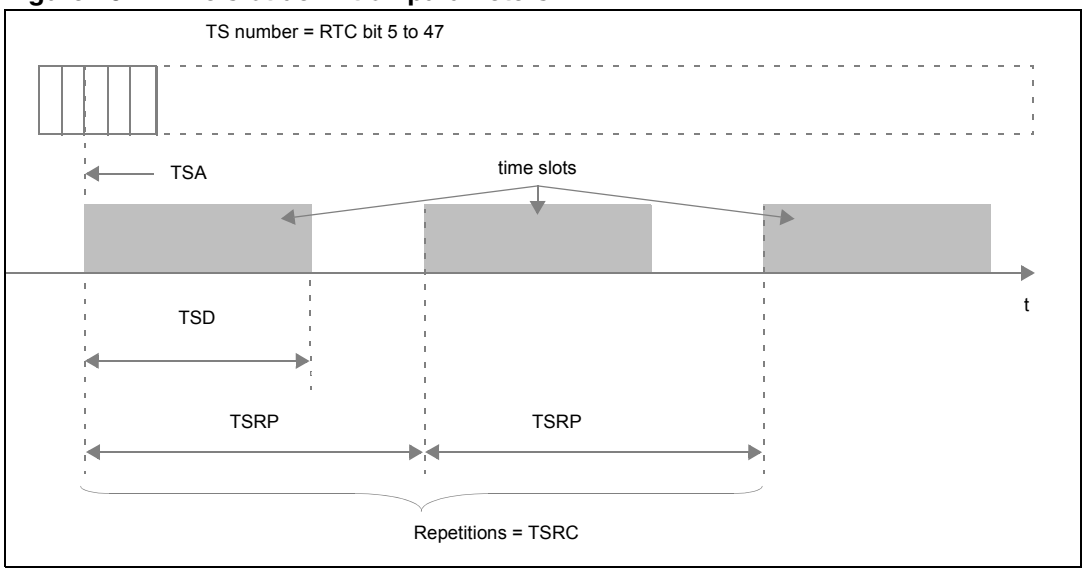
A time slot which has a length of one TSU allows the transmission of a small Data packet and the reception of the corresponding Ack packet (taking into account the RTC deviation between two stations over a certain time).

An individual time slot is identified by the following:

- Time Slot Identifier (TSID)
- Time Slot Alignment (TSA)
- Time Slot Duration (TSD)
- Time Slot Repetition Period (TSRP)
- Time Slot Repetition Count (TSRC)

Figure 76 illustrates these parameters.

Figure 76. Time slot definition parameters



The base unit of TSA, TSD and TSRP is a TSU. A TSU starts if bits 0 to 4 of the RTC are the value “0”.

Information about time slots are distributed by the master station to slave stations through the distribution of a Time Slot Table (TST). The master station transmits the TST in a Brdcast packet or forwards the TST to stations requesting it by an LCMP command.

Note: Time slot administration/negotiation/maintenance procedures are defined in Section 12.2: Link control management on page 66.

Only during an allocated time slot can a slave station transmit or receive messages. If a transmission exceeds the time slot, the transmission is aborted on both sides (transmitting and receiving stations) and an error is reported to the LLC.

To take into account the time deviation between the transmitting and receiving stations, the transmitting station must add a margin at the beginning of a time slot – a Time Slot Offset (TSO).

To avoid the reception of a packet in the wrong time slot or the rejection of a packet in the receiving station, the transmitting station can be configurable to delay the transmission of a packet. The delay can be configurable in 0 to 31 steps of 1/32,768 µs.

23.4.3 Logical channels

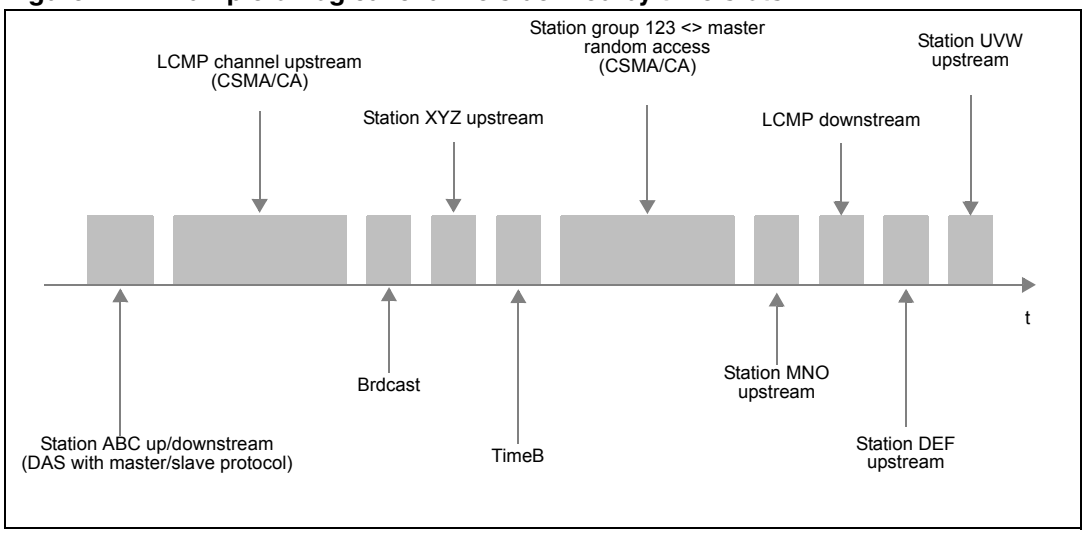
A master station can dedicate a time slot to:

- a slave station
- a group of slave stations
- all slave stations
- to itself
- to specific services

Specific services means that the time slot is used for specific functions, such as for the transmission of Brdcast, TimeB, or LCMP packets, as well as for specific application data.

In addition, time slots can be reserved for the direction of a transmission. An upstream slot is a time slot in which slave stations have access to the master station to transmit packets to it. A downstream slot is a time slot in which slave stations can access the master to transmit packets to (a) slave station(s).

Figure 77. Example of logical channels defined by time slots



Logical channels can also be used in DA mode or in CSMA/CA mode:

- In DA mode, a slot must either be dedicated to a single station or the master/slave protocol must be used.
- In CDMA/CA mode, multiple stations use a certain time slot for accessing the media in this mode (hybrid access).

23.4.4 Real-time clock

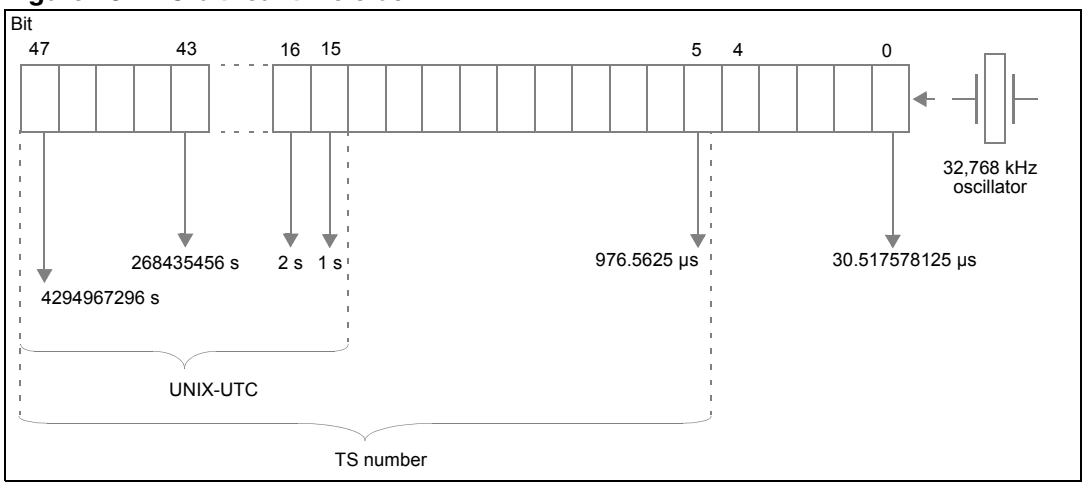
The real-time clock (RTC) is based on a modulo-2 48 bit counter and a 32,768 kHz oscillator frequency. It must be supported by all stations. The accuracy of the oscillator frequency is defined by ± 40 ppm. Consequently, the time deviation between stations over 1 second can reach 80 µs.

The time of the RTC in bits 15 to 47 is the number of seconds since the epoch, Jan. 1, 1970 00:00:00 (UNIX–UTC).

In a time slotted media, the master station maintains the reference clock of the network. A slave station then updates its RTC by obtaining a TimeB packet (a time beacon) from a master station. The master can transmit a TimeB packet in periodic intervals or the slave station can request a TimeB packet from the master through an LCMP command. The slave station is responsible for maintaining its RTC and for avoiding deviations. The master station defines the maximum deviation allowed in a network and transmits this definition via the TST. The master is also responsible for the time slot design and must take into account the maximum time deviation.

Figure 78 illustrates a 48-bit real-time clock.

Figure 78. 48-bit real-time clock



Note: See also Section 7.2: 32.786 kHz real-time clock (RTC) on page 27 for updating the real-time clock.

23.5 FDMA

To enable Networks to access the media at the same time the FDMA scheme can be used. All stations switched into the FDMA mode are using symbols with 22 MHz bandwidth. By using different carrier frequencies for each network these networks are separated by frequency band and thus are not disturbing each other.

FDMA is enabled by default in the TN100 transceiver.

23.5.1 Field for enabling FDMA

The following field is used to enable FDMA.

Table 48. Wake-up time fields

Field	Offset	R/W	Description
FdmaEnable	0x4A	WO brst BbClk	Enables FDMA (Frequency Division Multiple Access), which uses 22 MHz bandwidth. When disabled, 80 MHz bandwidth is used. Enabled by default.

Note: See Section 26.2.63: 0x4A – Baseband buffer and MACFrame configuration on page 172

24 Data transmission control methods

This section describes methods used in the TN100 transceiver to correct received data packets, filter out retransmissions, control message fragments, and manage unconfirmed data transmissions.

24.1 Symbol definitions

The following symbols are used in this section:

- SIFS
- MacSifsTime
- MacAckTimeMax
- MacArqCntMax

These symbols are defined in *Appendix A: Attributes and constants on page 221*.

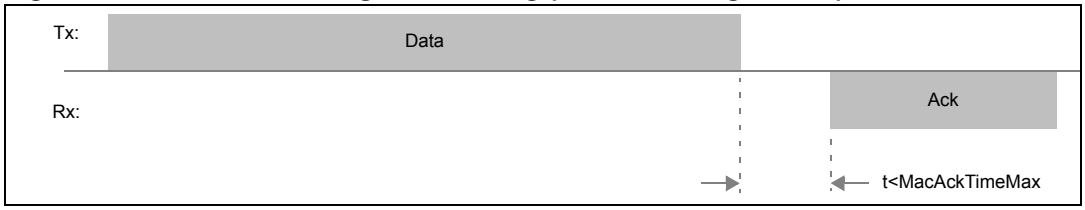
24.2 Error correction (ARQ scheme)

In addition to the FEC scheme, an Automatic Repeat Request (ARQ) scheme is used to achieve correct data transmission.

Note: *See Section 19.4: Forward error correction (FEC) on page 88.*

Upon receipt of a correct Data packet, a receiving station waits the SIFS (switching interframe space) interval, defined as MacSifsTime, and transmits an Ack packet (acknowledgment) back to the source station to indicate that the transmission was successfully receipt. The station transmitting the Data packet waits for the Ack packet for an interval, defined by constant MacAckTimeMax (24 µs), while taking into account the propagation delays of the air interface, as illustrated in *Figure 79*:

Figure 79. ARQ acknowledgement timing (at transmitting station)



Note: *MacAckTimeMax < 24 µs restricts usage of ARQ scheme to the stations which are located no more than . 2400 meters away from each other due to propagation delay of the air interface segment.*

If an Ack packet was not received at the end of a transmission, the source station retransmits the Data packet to the receiving station. To limit the number of unsuccessful attempts, only a maximum number of retransmissions are done as defined by the setting `MacArqCntMax`. A retransmit counter in the source station counts the number of unacknowledged transmissions.

If the source station receives a valid packet (such as a Data, Brdcast, TimeB, or Req2S packet) while it is waiting for the acknowledgement, it considers the transmission as unsuccessful. It then retransmits the Data packet after it processed the received packet (and

when the medium is idle again). If the source station receives an invalid packet (such as a Clr2S packet) while it is waiting for the acknowledgement, it is ignored and resumes retransmitting the Data packet (when the medium is again idle).

Note: For more details, see Section 23: Media access control methods on page 98.

To ensure that only a correct Data packet is accepted by a receiving station, the source station adds CRC codes (CRC1 and CRC2) to the packet. The receiving station can then perform one of two checks on that Data packet:

- The receiving station checks both CRC1 or CRC2. If either of these codes fail, an Ack packet is not transmitted.
- The receiving station checks only CRC1. If CRC1 succeeds, then an Ack packet is then transmitted, irrespective of the CRC2. This mode can be used if it is unnecessary to ensure the integrity of the data in the Data field of the MACFrame (for example, in voice applications).

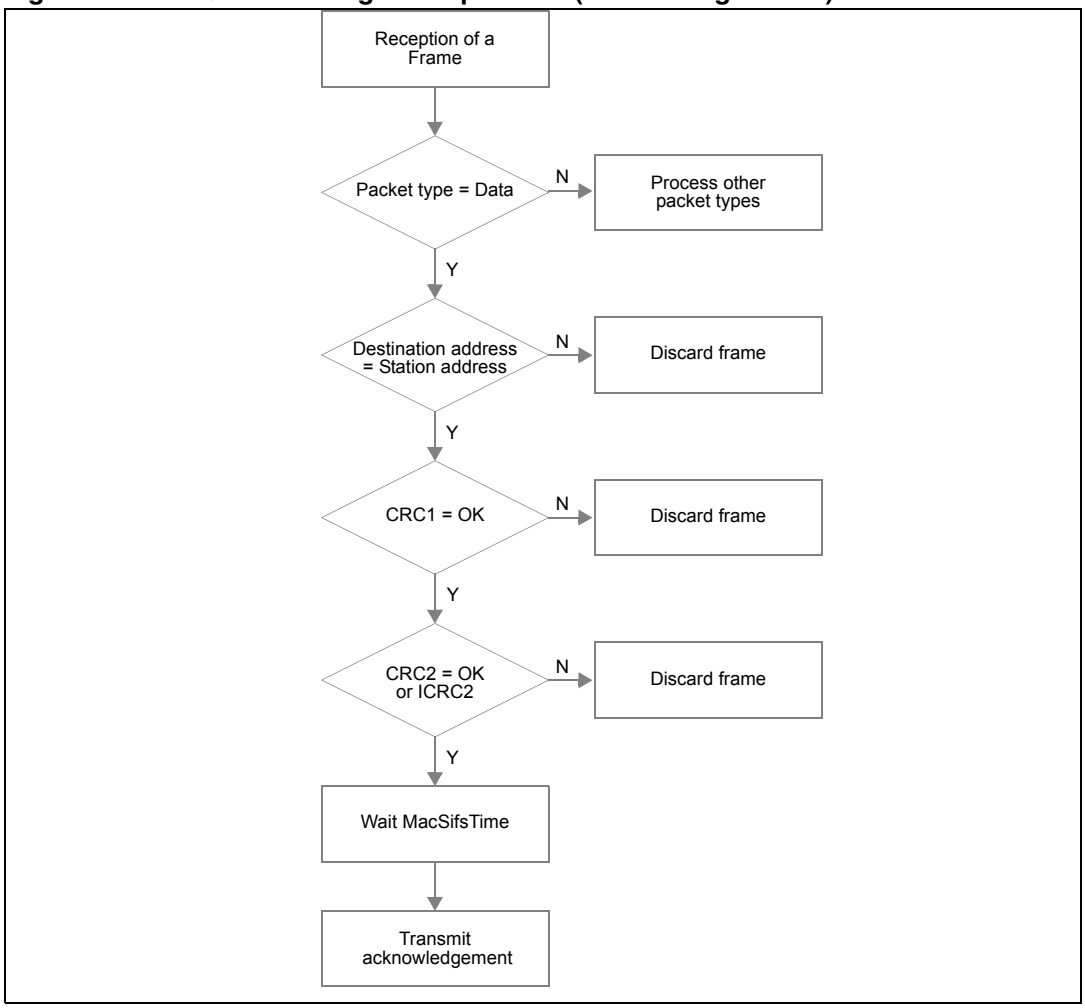
A station is configurable to select both modes.

Note: For more details, see Section 19.2: Cyclic redundancy check (CRC) on page 86.

Data packets are acknowledged by the addressed station only. Brdcast and TimeB must not be acknowledged since these packets are destined for more than one station.

The protocol for transmitting an acknowledgement to a received packet is illustrated in *Figure 80*.

Figure 80. ARQ acknowledgement protocol (at receiving station)



24.3 Retransmit filtering

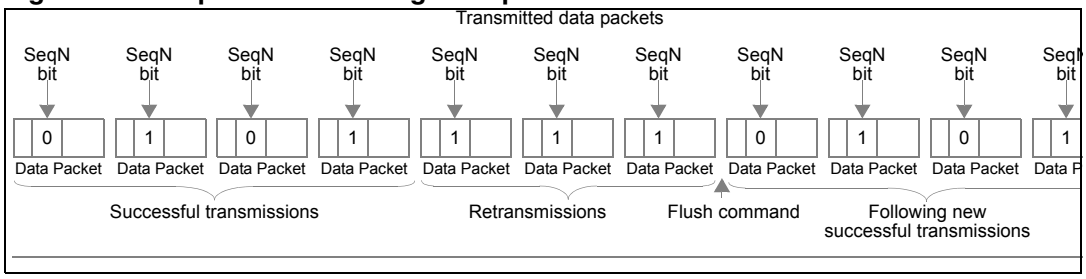
The SeqN bit of the FrameControl field identifies retransmitted packets so that the receiver can discard duplicates of successfully received Data packets.

The SeqN bit provides a sequential numbering scheme to order the Data packet stream. For each new transmitted packet, the SeqN bit is inverted. This is required to filter out retransmissions at the destination. For instance, if a retransmission occurs due to a failing acknowledge, the destination will receive the same packet twice. By comparing the SeqN of consecutive packets, correctly received retransmissions can be discarded.

A Data packet is retransmitted until either a positive acknowledgment is received, the MacArqCntMax attribute is reached, or a time-out is exceeded. A retransmission is carried out because the packet transmission itself failed or because the acknowledgment failed. In the latter case, the destination keeps receiving the same Data packet over and over again.

To filter out the retransmissions caused by failed acknowledgements at the destination, a SeqN bit is added to the FrameControl field of the MACFrame. Normally, this bit is alternated for every new Data packet transmission, except during retransmissions, as shown in *Figure 81*.

Figure 81. Sequence numbering example



Successfully transmitted Data packets alternate the value of the SeqN bit between “0” and “1”, except for retransmissions. During retransmissions, the last value of the SeqN bit is retained until a maximum retransmission value is reached, in which case, the transmitter performs a flush command and then waits for software to provide another Data packet for transmission.

Retransmission filtering is performed as follows:

1. On the sending station, the transmitter sends a Data packet and waits for an acknowledgement.
2. when an Ack packet is received successfully the SeqN bit is inverted and a new Data packet is sent.
3. If no Ack packet has been received, the Data packet is retransmitted with the same SeqN number.
4. If the MacArqCntMax attribute is reached or a time-out is exceeded, the transmitter must flush the unacknowledged Data packet to be ready to transmit a new Data packet. The flush is done by a LCMP (Link Control Management Protocol) command. This procedure prevents the loss of a Data packet due to the retransmit filtering.
5. On the destination station, the receiver compares the SeqN bit with the previous SeqN value. If they are different, a new Data packet has arrived; otherwise it is the same Data packet and can be discarded.

For more details, see *Section 12.2: Link control management on page 66*.

Since the ARQ scheme is only applied to Data packets, the SeqN bit in a Brdcast packet is not meaningful. This bit is, therefore, in this case set to the value “0”.

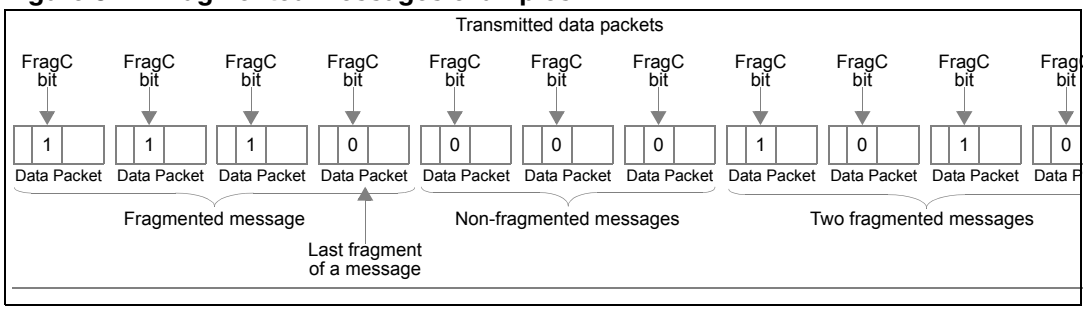
24.4 Fragmentation control

The FragC bit of the FrameControl field identifies fragmented messages and is used to support the fragmentation of messages. A value of “1” in this field indicates that at least one additional fragment will follow the packet containing this fragment. A value of “0” indicates the last fragment or the frame is not fragmented.

A message can be fragmented into several Data packets. The FragC bit of a Data packet set to the value “1” indicates that the packet is carrying either the first fragment or the continuing fragments of the message. The FragC bit set to the value “0” indicates that either the Data packet is the last fragment of a fragmented message or that the message has not been fragmented. The FragC bits in non-fragmented messages are always set to the value “0”.

Examples of fragmented messages are shown in *Figure 82*.

Figure 82. Fragmented messages examples



24.5 Unconfirmed data transmission

A station is configurable to transport packets in an entirely non-confirmed way.

The station must not transmit an Ack packet (an acknowledgment) in response to a successfully received Data packet. This mode is not applicable to a three-way handshake. For more details, see *Section 23.3.5: Three-way handshake on page 102*.

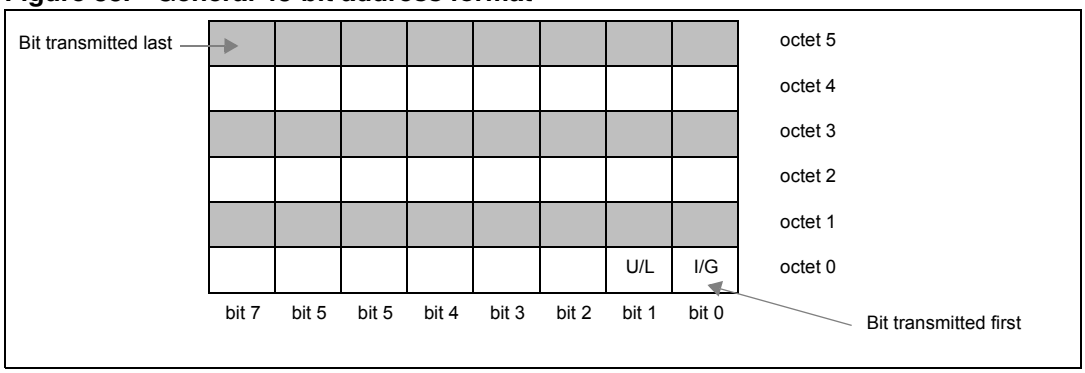
25 Address formats

This section describes addressing and message types in the TN100 transceiver.

25.1 General address format / addressing

The general address size is application specific. If 48 bit addresses are used, then the convention can follow the IEEE Std 802-1990 standards for LAN and MAN networks.

Figure 83. General 48-bit address format



The 48 bit address[c] is represented as a string of six octets. These octets are displayed from left to right in the order in which the are transmitted, separated by hyphens. Each octet is displayed as two hexadecimal digits. The bits within the octets are transmitted with LSB first.

A 48 bit address consists of two parts:

- The first 24 bits correspond to the Organizationally Unique Identifier as assigned by the IEEE with the exception that the first bit may be set by the assignee to either “1” for group addresses or “0” for individual addresses.
- The second and remaining 24 bits is administered locally by the assignee.

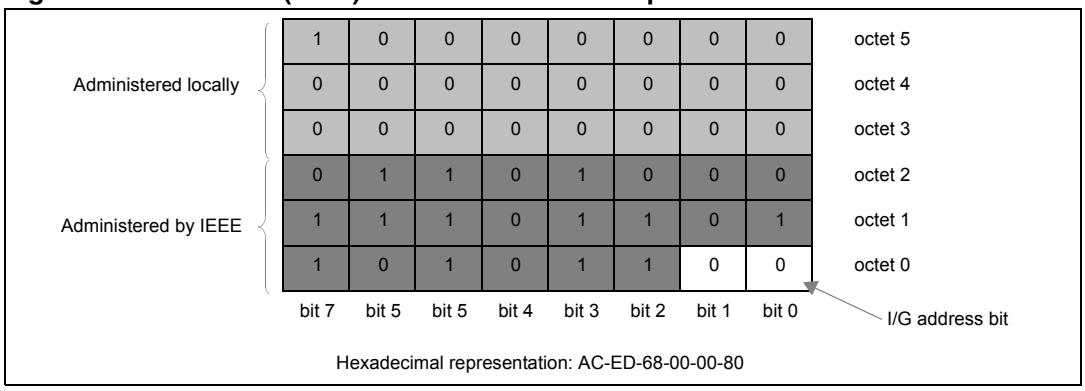
Bit 0 of octet 0 is the I/G Address Bit. It is used to identify the destination address as either an individual or as a group. The I/G Address Bit can be set to either “0” or “1”:

- “0” indicates that the address field contains an individual address. This is a unicast message, where “0” = I.
- ”1” indicates that the address field contains a group address that identifies one, many, or all stations. This is a multicast message, where “1” = G.

Note: A special, predefined group address of all 1’s is an all-stations broadcast address for broadcast messages.

c. Since the first two bits are not available for the address space, only 46 bits are actually available. This results in 2^{46} = 70,368,744,177,664 different addresses.

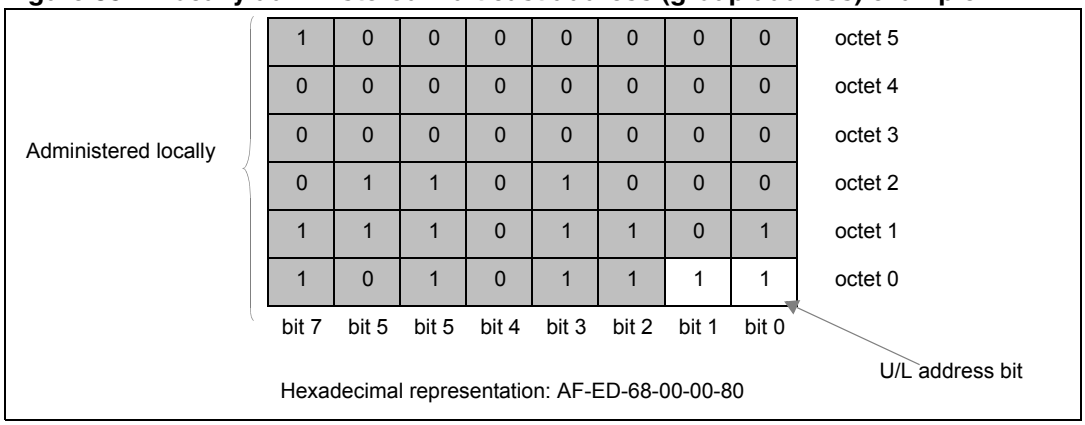
Figure 84. Universal (IEEE) unicast address example



Bit 1 of octet 0 is the U/L Address Bit. It is used to identify the address as either universally or locally administrated. The U/L Address Bit can be set to either “0” or “1”:

- “0” indicates that the addresses are universally administered, where “0” = U.
- “1” indicates that the entire address (all 48 bits) has been locally administered, where “1” = L.

Figure 85. Locally administered multicast address (group address) example



25.2 Addressing

- **Broadcast messages**

  In a broadcast message, all bits of the Address1 field in a Brdcast packet are set to the value “1”. If a Brdcast packet is received with the I/G bit set to “0”, which would indicate a unicast message, it is discarded without indication to the LLC.

- **Multicast messages**

  In a multicast message, bit 0 of octet 0 is set to the value “1” while bit 1 of octet 0 identifies it as either a local or a universal (IEEE) address. The remaining bits represent a group address.

- **Unicast messages**

  In a unicast message, bit 0 of octet 0 is set to the value “0”. Since a Data packet is always a unicast message, its I/G bit must be set to “0”. If a Data packet is received with the I/G bit set to “1”, which would indicate a multicast message, it is discarded without indication to the LLC.

- **Promiscuous mode**

  When a station is set to promiscuous mode, it is able to receive Data packets, which are unicast messages, addressed to any station and report them to the LLC. However, if the destination address does not match the station address, it must not send an acknowledgement.

- **User-defined address conventions**

  Since the distinction between a unicast, multicast, or broadcast message is accomplished by the I/G bit, coding a Brdcast packet through the TypeCode field is redundant. However, this redundancy allows the application of user-defined address conventions that are independent of this convention.

26 Chip registers

26.1 Chip register mapping

Shaded fields are ST Reserved.

Table 49. Memory map of chip registers: 0x00 – 0x7F

Offset	RW	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00	R					Irq Driver	Irq Polarity	SpiTx Driver	SpiBit Order
	W					Irq Driver	Irq Polarity	SpiTx Driver	SpiBit Order
0x01	R	Version							
	W	WakeUpTimeByte							
0x02	R	Revision							
	W					WakeUpTimeWe			
0x03	R	BattMgmt Enable			BattMgmt Threshold				BattMgmt Compare
	W	BattMgmt Enable			BattMgmt Threshold				
0x04	R					DioInValueAlarmStatus			
	W			DioUse Pulldown	DioUse Pullup	DioAlarm Polarity	DioAlarm Start	DioOut Value Alarm Enable	Dio Direction
0x05	R								
	W					DioPortWe			
0x06	R	Power Down Mode	PowerUpTime					Enable Wake UpDio	Enable Wake UpRtc
	W	Power Down Mode	PowerUpTime					Enable Wake UpDio	Enable Wake UpRtc
0x07	R						Reset BbRadio Ctrl	Reset BbClock Gate	
	W	ST Reserved	ST Reserved				Reset BbRadio Ctrl	Reset BbClock Gate	Power Down
0x08	R	Enable Feature Clock	FeatureClockFreq				Bypass BbCrystal	Enable BbClock	Enable BbCrystal
	W	Enable Feature Clock	FeatureClockFreq				Bypass BbCrystal	Enable BbClock	Enable BbCrystal
0x09	R								
	W	Use Pulldown 4Spitxd	Use Pullup 4Spitxd	Use Pulldown 4Spirxd	Use Pullup 4Spirxd	Use Pulldown 4Spissn	Use Pullup 4Spissn	Use Pulldown 4Spiclk	Use Pullup 4Spiclk

Table 49. Memory map of chip registers: 0x00 – 0x7F (continued)

Offset	RW	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0A	R								
	W	Use Pulldown 4Ucrst	Use Pullup 4Ucrst	Use Pulldown 4Ucirq	Use Pullup 4Ucirq	Use Pulldown 4Pamp	Use Pullup 4Pamp	Use Pulldown 4Por	Use Pullup 4Por
0x0B	R								
	W							Write Pulls 4Pads	Write Pulls 4Spi
0x0C	R					ST Reserved			
	W					ST Reserved			
0x0D	R					ST Reserved			
	W					ST Reserved			
0x0E	R			DeviceSelect				RamIndex	
	W			DeviceSelect				RamIndex	
0x0F	R	LoIrq Status	BbTimer IrqStatus	RxIrq Status	TxIrq Status	LoIrq Enable	BbTimer IrqEnable	RxIrq Enable	TxIrq Enable
	W					LoIrq Enable	BbTimer IrqEnable	RxIrq Enable	TxIrq Enable
0x10	R			TxIntsRawStat					
	W			TxIntsReset					
0x11	R		RxIntsRawStat						
	W		RxIntsReset						
0x12	R							LoInts RawStat	ST Reserved
	W	Clear Base band TimerInt						LoInts Reset	ST Reserved
0x13	R								
	W			TxIntsEn					
0x14	R								
	W		RxIntsEn						
0x15	R								
	W							LoIntsEn	ST Reserved
0x16	R	LoRxCapsValue							
	W	LoRxCapsValue							
0x17	R	LoRxCapsValue							
	W	LoRxCapsValue							
0x18	R			LoRxCapsValue					
	W			LoRxCapsValue					
0x19	R	LoTxCapsValue							
	W	LoTxCapsValue							
0x1A	R	LoTxCapsValue							
	W	LoTxCapsValue							

Table 49. Memory map of chip registers: 0x00 – 0x7F (continued)

Offset	RW	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x1B	R			LoTxCapsValue					
	W			LoTxCapsValue					
0x1C	R								
	W	UseLo RxCaps		ST Reserved	LoEnable LsbNeg	LoFastTuningLevel			LoEnable FastTuning
0x1D	R	LoTargetValue							
	W	LoTargetValue							
0x1E	R	LoTargetValue							
	W	LoTargetValue							
0x1F	R								
	W	AgcThresHold1							
0x20	R								
	W	AgcThresHold2							
0x21	R								
	W	HoldAgc InFrame Sync	HoldAgcInBitSync						
0x22	R								
	W	AgcNregLength		ST Reserved					
0x23	R								
	W	AgcIntTime							
0x24	R								
	W					AgcIntTime			
0x25	R								
	W	AgcHold	Agc DefaultEn	AgcValue					
0x25	R								
	W	AgcValue							
0x26	R			AgcGain					
	W	ST Reserved		AgcGainThres					
0x27	R			FctPeriod					
	W	EnableTx		StartFct Measure	FctClock En	ChirpFilterCaps			
0x28	R								
	W	BasebandTimerStartValue							
0x29	R								
	W	BasebandTimerStartValue							
0x2A	R	ToaOffsetMeanAck							
	W	SyncWord							
0x2B	R	ToaOffset Mean AckValid			ToaOffsetMeanAck				
	W	SyncWord							

Table 49. Memory map of chip registers: 0x00 – 0x7F (continued)

Offset	RW	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x2C	R	TxRespTime							
	W	SyncWord							
0x2D	R	TxRespTime							
	W	SyncWord							
0x2E	R		PhaseOffsetData				PhaseOffsetAck		
	W	SyncWord							
0x2F	R	ToaOffsetMeanData							
	W	SyncWord							
0x30	R	ToaOffset Mean DataValid			ToaOffsetMeanData				
	W	SyncWord							
0x31	R	RxCrc2 Stat	RxCrc1 Stat	RxAddrMatch		RxPacketType			
	W	SyncWord							
0x32	R					RxCorrBitErr			
	W	RxCorrErrThres							
0x33	R	RxCrypt SeqN	RxCryptId		RxCrypt En			RxAddr Seg IsMatch	RxAddr Seg EsMatch
	W	TxTimeSlotStart							
0x34	R	RxFec1BitErr							
	W	TxTimeSlotStart							
0x35	R		RxFec1BitErr						
	W	TxTimeSlotEnd							
0x36	R								
	W	TxTimeSlotEnd							
0x37	R								
	W							RxTime Slot Control	TxTime Slot Control
0x38	R	RxPacketSlot							
	W	RxTimeSlotStart							
0x39	R	RxPacketSlot							
	W	RxTimeSlotStart							
0x3A	R								
	W	RxTimeSlotEnd							
0x3B	R								
	W	RxTimeSlotEnd							
0x3C	R					TxArqCnt			
	W	TxArqMax							

Table 49. Memory map of chip registers: 0x00 – 0x7F (continued)

Offset	RW	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x3D	R								
	W	ST Reserved	Csq UseRam	Csq Mem AddrInit	Csq AsyMode	Csq Use 4Phases	CsqUse Phase Shift	CsqDitherValue	
0x3E	R								
	W	ST Reserved							
0x3F	R								
	W	ST Reserved				D3lPomLen		D3lPom En	D3lFixn Map
0x40	R								
	W	UseMap Thresh1 InFrame sync	LeaveMapThresh1InBitsync						
0x41	R								
	W	ST Reserved	Go2MapThresh1InBitsync						
0x42	R								
	W	ST Reserved			Enable ExtPA	Invert RxClock	Enable CsqClock	Enable LOdiv10	Enable LO
0x43	R								
	W		TxPaBias				LnaFreqAdjust		
0x44	R								
	W			TxOutputPower0					
0x45	R								
	W			TxOutputPower1					
0x46	R								
	W				RfRxCompValueI				
0x47	R								
	W				RfRxCompValueQ				
0x48	R								
	W	Modulation System	SymbolRate				SymbolDur		
0x49	R								
	W	TxRxCryptClkMode				TxRx CryptCrc2 Mode	Use Fec	Crc2Type	
0x4A	R								SwapBb Buffers
	W	TxRx Mode			Fdma Enable		TxRxBb Buffer Mode0	TxRxBb Buffer Mode1	SwapBb Buffers
0x4B	R								
	W		ChirpMatrix1				ChirpMatrix0		

Table 49. Memory map of chip registers: 0x00 – 0x7F (continued)

Offset	RW	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x4C	R								
	W		ChirpMatrix3				ChirpMatrix2		
0x4D	R								
	W	TxMac CifsDis			TxUnder runIgnore	TxPreTrailMatrix1		TxPreTrailMatrix0	
0x4E	R								
	W	TxFrag Prio	TxBack OffAlg	Tx3Way	TxArq	TxVCarr SensAck	TxPhCarrSenseMode		TxVCarr Sens
0x4F	R								
	W	TxBackOffSeed							
0x50	R								
	W	TxCrypt SeqN	TxCryptId		TxCrypt En	TxCryptSeqReset			
0x51	R								
	W	TxScramb En	TxScrambInit						
0x52	R								
	W	TxTransBytes							
0x53	R								
	W				TxTransBytes				
0x54	R								
	W	TxAddr Slct				TxPacketType			
0x55	R								
	W					TxBufferCmd		TxCmd Start	TxCmd Stop
0x56	R								
	W					RxBufferCmd		RxCmdSta rt	RxCmdSto p
0x57	R								
	W					RxCryptSeqReset			
0x58	R								
	W	RxTransBytes							
0x59	R								
	W				RxTransBytes				
0x5A	R								
	W	RxAddrSegDevIdL		RxAddr Seg IsMode	RxAddr Seg EsMode	RxArqMode		RxCrc2 Mode	RxTimeB Crc1 Mode
0x5B	R								
	W	RangingPulses				Rx Addr Mode	Rx TimeBEn	Rx Brdcast En	RxDataEn
0x5C	R								
	W				PulseDetDelay				

Table 49. Memory map of chip registers: 0x00 – 0x7F (continued)

Offset	RW	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x5D	R								
	W			Up Pulse DetectDis	Down Pulse DetectDis		GateAdjThreshold		
0x5E	R								
	W	GateAdj Frame syncEn	GateAdj BitsyncEn	GateSizeFramesync		GateSizeBitsync		GateSizeUnsync	
0x5F	R								
	W		LeaveBitsyncThreshold				Go2BitsyncThreshold		
0x60	R								
	W	RtcTimeBTxAdj							
0x61	R								
	W	RtcTimeBRxAdj							
0x62	R								
	W	ST Reserved			Rtc TimeB AutoMode			Rtc CmdRd	RtcCmdWr
0x63	R								
	W				AgcAmplitude				
0x64	R								
	W	UseAlter nativeAgc				AgcRangeOffset			
0x7D	R					ST Reserved			ST Reserved
	W	ST Reserved							ST Reserved
0x7E	R	ST Reserved				ST Reserved			
	W								
0x7F	R	ST Reserved		ST Reserved	ST Reserved	ST Reserved			
	W								

26.2 Description of chip registers

This section describes the 128 byte programmable register space for chip configuration settings.

Table 50 lists the conventions that apply to the *Properties* (Prop.) column of the field description tables.

Table 50. Field properties column

Convention	Description
R	This field supports read only mode.
W	This field supports write only mode.
RW	This field supports both read and write mode.
brst	This field supports SPI burst access and can be a located within a block that is read from or written to in a burst.
strb	This field supports a strobe function where only a single write access is required for generating a command or action. The value `True` must be written to the location only once.
BbClk	To access this field, the baseband clock distribution must be enabled.
SPI	Field runs with the SPI clock.

26.2.1 0x00 – SPI bit order and IRQ pad configuration

Used for selecting the SPI bit order and for configuring the SPI TX and IRQ pads.

Address offset: 0x00

R/W	7	6	5	4	3	2	1	0
RW					Irq Driver	Irq Polarity	SpiTx Driver	SpiBit Order
init	0	0	0	0	0	0	0	0

Table 51. Field properties for register 0x00

Bits	Description
0	**SpiBitOrder**: *Select MSB (Most Significant Bit) or LSB (Least Significant Bit) Bit Order for SPI* Defines the bit order of the SPI. However, the byte order always remains the same. – NA_SpiBitOrderLsbFirst_C (0x0) The first bit transmitted over the SPI interface is the LSB, while the last bit is the MSB. Each byte is transmitted with LSB first. – NA_SpiBitOrderMsbFirst_C (0x1) The first bit transmitted over the SPI interface is the MSB, while the last bit is the LSB. Each byte is transmitted with MSB first. Note: After PowerUpReset or after wakeup from PowerDownModeFull, the SPI is configured as LSB first. For more information, see *Section 26.2.2: Configuring SPI transfers when the bit order is unknown on page 126*.

Table 51. Field properties for register 0x00 (continued)

Bits	Description
1	**SpiTxDriver**: *Select push-pull or open-drain for SPI TxD Output Driver* Switches between push-pull and open-drain for the TxD output driver (SpiTxD - pin 17). – NA_SpiTxDriverOpenDrain_C (0x0) The SpiTxD output driver is open-drain. The pad is driven only when a logic 0 is sent from TN100 to the SPI master. Otherwise, the output is in high-impedance state. – NA_SpiTxDriverPushPull_C (0x1) The SpiTxD output driver is push-pull. The pad is driven only when data is sent from TN100 to the SPI master. Otherwise, the output is in high-impedance state.
2	**IrqPolarity**: *Select High or Low Active for IRQ Polarity* Defines the polarity (high or low active) of the interrupt request signal. – NA_IrqPolarityActiveLow_C (0x0) IRQ is low active. – NA_IrqPolarityActiveHigh_C (0x1) IRQ is high-active.
3	**IrqDriver**: *Select push-pull or open-drain for IRQ Driver* Switches between the push-pull and the open-drain driver for the IRQ output driver (µCIRQ - pin 27). – NA_IrqDriverOpenDrain_C (0x0) The IRQ output driver is open-drain. – NA_IrqDriverPushPull_C (0x1) The IRQ output driver is push-pull.

26.2.2 Configuring SPI transfers when the bit order is unknown

Because only the lower half of the register is used, the configuration of the SPI transfer is always possible even when the bit order for the TN100 transceiver is unknown. This can be easily seen, for instance, by looking at the first two bytes (the Instruction and Address bytes) and at the one-byte write transfer to the address 0x00.

Note: See also Section 9: Programming interface (SPI) on page 42.

The Instruction and Address bytes are always 0x81 0x0. This means:

- 1st byte (Instruction field): write one byte
- 2nd byte (address field): select address 0

The same value is received when bit order is wrong – mirroring 0x81 leads to 0x81 as well. The same is true for the third byte of the transfer. For bits 0 to 3, mirror these bits to bits 7 to 4 (for example, 0 to 7; 1 to 6; 2 to 5; and 3 to 4). The following table provides four examples where it is possible to set the bit order of the registers and configure bits 1 to 3, even if the bit order already set in the chip is unknown.

Table 52. Configuring the TN100 when bit order of chip is unknown

Configuring MSB First	Configuring LSB First	Configuration of Bits 1 to 3		
0x81 0x0 0x81	0x81 0x0 0x0	SpiTxDriver = Open Drain	IrqPolarity = Active Low	IrqDriver = Open Drain
0x81 0x0 0xDB	0x81 0x0 0x5A	SpiTxDriver = Push Pull	IrqPolarity = Active Low	IrqDriver = Push Pull
0x81 0x0 0xE7	0x81 0x0 0x66	SpiTxDriver = Push Pull	IrqPolarity = Active High	IrqDriver = Open Drain
0x81 0x0 0x99	0x81 0x0 0x18	SpiTxDriver = Open Drain	IrqPolarity = Active Low	IrqDriver = Push Pull

26.2.3 0x01 – Digital controller version number and wake-up time byte

Used for reading the version number of the digital controller as well as for writing a wake-up time value of one byte. This value is used with `WakeUpTimeWe` to program the wake-up time.

Note: See Section 26.2.3: 0x01 – Digital controller version number and wake-up time byte on page 127.

Address offset: 0x01

RW	7	6	5	4	3	2	1	0
R	Version							
init	0	0	0	0	0	1	0	1
W	WakeUpTimeByte							
init	0	0	0	0	0	0	0	0

Table 53. Field properties for register 0x01

Bits	Prop	Description
0-7.	RO	**Version**: *Version Number of Digital Controller* Default value: 0x5
0-7	WO	**WakeUpTimeByte**: *Real-time clock Wake-Up Time Byte* This field stores a one byte value for the wake-up time, which is programed to the wake-up time RtcWakeUpTime using WakeUpTimeWe. The complete programmed wake-up time is compared to the bits 31:8 of the real-time clock. The LSB of the programmed wake-up time corresponds to a 7812.5 µs cycle (bit 8 of the RTC). RtcWakeUpTime can only be accessed through WakeUpTimeByte and WakeUpTimeWe. See *Section 26.2.4: 0x02 – Digital controller revision number and wake-up time on page 127*.

26.2.4 0x02 – Digital controller revision number and wake-up time

Used for reading the revision number of the digital controller as well as for writing the value of `WakeUpTimeByte` to the wake-up time circuitry.

Note: See Section 26.2.3: 0x01 – Digital controller version number and wake-up time byte.

Address offset: 0x02

RW	7	6	5	4	3	2	1	0
R	Revision							
init	0	0	0	0	0	0	0	1
W					WakeUpTimeWe			
init	0	0	0	0	0	0	0	0

Table 54. Field properties for register 0x02

Bits	Prop	Description
0-7	RO	**Revision**: *Revision Number of Digital Controller* Default value: 0x1
1-3	WO brst SPI	**WakeUpTimeWe**: *Wake-Up Time Byte Selector* Loads the value of RtcWakeUpTimeByte to the appropriate byte of the wake-up time in the wake-up time circuitry. Default value: 0x0 A 1-0 sequence must be written to this field before changing RtcWakeUpTimeByte. See also *Section 26.2.3: 0x01 – Digital controller version number and wake-up time byte on page 127*.

26.2.5 0x03 – Battery management

Used for enabling the voltage comparator function, for setting the voltage comparator value, and for reporting the voltage status of the battery.

Note: The battery management function is completely realized in the analog part of the chip.

Address offset: 0x03

RW	7	6	5	4	3	2	1	0
R	BattMgmt Enable			BattMgmtThreshold				BattMgmt Compare
W								

Table 55. Field properties for register 0x03

Bits	Mnemonic	Prop	Description
1-4		RW brst SPI	**BattMgmtThreshold**: *Battery Management Threshold* Sets battery comparator level. These bits set the comparison voltage. The chip checks if the compare value is higher or lower than the real battery voltage. *Value* *Voltage* 0000 2.15V 0001 2.19V 0010 2.23V 0011 2.27V 0100 2.31V 0101 2.35V 0110 2.39V 0111 2.43V 1000 2.47V 1001 2.51V 1010 2.55V 1011 2.59V 1100 2.63V 1101 2.67V 1110 2.71V 1111 2.75V Default value: 0x0 The BattMgmtCompare bit indicates the result of the comparison.
7		RW brst SPI	**BattMgmtEnable**: *Enable Battery Management* Enables the battery voltage comparator function. 0: Disable (Default) 1: Enable Must be enabled before BattMgmtCompare can be read.
0		RO	**BattMgmtCompare**: *Battery Comparator Output* Reports if the battery voltage is above or below the value of BattMgmtThreshold. 0: Below 1: Above The read value is valid only if the Battery Comparator is enabled, that is, when BattMgmtEnable = 1.

26.2.6 0x04 – Digital I/O controllers and digital I/O alarm status

The digital I/O ports (D0 to D3, pins 19 to 23) can be used as a normal input or to report the occurance of an alarm. When used as a normal input, the signal level at these ports can be read. When used as a normal output, a programmable value can be driven out of the chip. When used to report the occurance of an alarm, They could be used to wake-up the chip. A clock (feature clock) could then be driven out of the chip.

When reading this register, if set as a normal input each bit reports the signal level of the corresponding digital IO port. Otherwise, each bit reports the occurrence of an alarm at the

corresponding digital IO port. When writing to this register, the values are set inside the register. The values first influence one or more digital IO ports when a write strobe is generated for the desired digital IO port(s) via register 0x05 (see *Section 26.2.7: 0x05 – Write enable digital I/O port*). This causes the values in this register to be copied to the corresponding configuration registers of these digital IO ports.

Address offset: 0x04
Reset value: 0x00

RW	7	6	5	4	3	2	1	0
R					DioInValueAlarmStatus			
W			DioUse Pulldown	DioUse Pullup	DioAlarm Polarity	DioAlarm Start	DioOutValue AlarmEnable	Dio Direction

Table 56. Field properties for register 0x04

Bits	Prop	Description
0-3	RO	**DioInValueAlarmStatus**: *Signal Level / Alarm Status* Each bit reports the signal level or the occurrence of an alarm at one of the four digital IO ports. – Bit 0 belongs to D0 – Bit 1 belongs to D1 – Bit 2 belongs to D2 – Bit 3 belongs to D3 Interpreting the bits make sense only when the corresponding digital IO port is set as an input (DioDirection = 0). When the digital IO port is used as a normal input (DioOutValueAlarmEnable = 0), then the read value is the signal level at that port. When the digital IO port is used to report the occurrence of an alarm (DioOutValueAlarmEnable = 1), then reading a 1 means that an alarm has occurred.
0	WO brst SPI	**DioDirection**: *Control Direction (Input or Output) of a Digital IO Port* 0: Digital IO port is configured as an input 1: Digital IO port is configured as an output
1	WO brst SPI	**DioOutValueAlarmEnable**: *Set Digital IO as either an Alarm Input or as a Value* When a digital IO port is set as an input [DioDirection = 0], this bit reports either the signal level at that port or the occurrence of an alarm at that port. When the digital IO port is set as an output (DioDirection = 1), the value programmed is driven out of the port. 0: Normal input / value to drive out of port 1: Enable alarm / value to drive out of port
2	WO brst SPI	**DioAlarmStart**: *Start Digital IO Alarm* This bit starts the alarm functionality. It must be set after the digital IO port is set as an input (DioDirection = 0) and is set to report the occurrence of an alarm (DioOutValueAlarmEnable = 1), 0: False 1: True (start alarm)

Table 56. Field properties for register 0x04 (continued)

Bits	Prop	Description
3	WO brst SPI	**DioAlarmPolarity**: *Set Digital IO Alarm Polarity /Source to Drive Digital IO Port* When the digital IO port is set as an input (DioDirection = 0) and should report the occurrence of an alarm (DioOutValueAlarmEnable = 1), then this bit is used to select the edge which should trigger the alarm. When the digital IO port is set as an output (DioDirection = 0) then this bit selects whether the value programmed in DioOutValueAlarmEnable or the feature clock should be driven out of the digital IO port. 0: Triggers alarm on falling edge / drive value programmed in DioOutValueAlarmEnable 1: Triggers alarm on rising edge / drive feature clock

26.2.7 0x05 – Write enable digital I/O port

Used for generating a write strobe for the configuration registers of the digital IO ports (D0 to D3, pins 19 to 23) by software. When the write strobe occurs for one digital IO port, the values programmed in register 0x04 are copied into the corresponding registers of that digital I/O. See *Section 26.2.6: 0x04 – Digital I/O controllers and digital I/O alarm status*. When more than one digital IO port should be configured with the same values, the write strobe for these digital IO ports can be programmed in parallel. The bit position in this register determines to which digital IO port configuration is written:

- 0x1 → Apply configuration to D0
- 0x2 → Apply configuration to D1
- 0x4 → Apply configuration to D2
- 0x8 → Apply configuration to D3
- 0x3 → Apply configuration to D0 and D1
- and so on

A write strobe must always be generated by programming a 1-0-sequence to this register. Do NOT change the content of register 0x04 *0x04 – Digital I/O controllers and digital I/O alarm status* when one of the bits of this register is active.

Address offset: 0x05
Reset value: 0x00

RW	7	6	5	4	3	2	1	0
R								
W					DioPortWe			

Table 57. Field properties for register 0x05

Bits	Prop	Description
0-3	WO brst SPI	**DioPortWe**: *Write Enable a Digital IO Port* Writes the settings of the following fields to the fields in the digital IO controller: – DioDirection – DioOutValueAlarmEnable – DioAlarmPolarity – DioAlarmStart – DioUsePullup – DioUsePulldown A 1-0 sequence must be written to this field before changing register 0x04. Writing a 1-0 sequence to one (or more) bits of DioPortWe writes the value of 0x4 into the registers of the associated digital IO pin(s).

26.2.8 0x06 – Power management

Used for selecting the type of power down mode, for configuring the power-up time, and for defining the event that brings the chip into the power-up state.

Address offset: 0x06
Reset value: 0x00

RW	7	6	5	4	3	2	1	0
RW	PowerDown Mode	PowerUpTime					Enable WakeUpDio	Enable WakeUpRtc

Table 58. Field properties for register 0x06

Bits	Prop	Description
0	RW brst SPI	**EnableWakeUpRtc**: *Enable real-time clock as Wake-Up Source* Enables the real-time clock as a wake-up source. It bring the chip into the power up state. 0: Disable 1: Enable
1	RW brst SPI	**EnableWakeUpDio**: *Enable Digital IOs as Wake-Up Source* Enables an alarm event (rising/falling edge) at one of the digital IOs configured as a wake-up source. It brings the chip into the power up state. See also *Section 26.2.6: 0x04 – Digital I/O controllers and digital I/O alarm status on page 129*. 0: Disable 1: Enable

Table 58. Field properties for register 0x06 (continued)

Bits	Prop	Description
4-6	RW brst SPI	**PowerUpTime**: *Configure Power Up Time* Configures the duration of the power-up time (duration of active µCReset). – PowerUpTime128Ticks_C (0x0) – PowerUpTime1Ticks_C (0x1) – PowerUpTime2Ticks_C (0x2) – PowerUpTime4Ticks_C (0x3) – PowerUpTime8Ticks_C (0x4) – PowerUpTime16Ticks_C (0x5) – PowerUpTime32Ticks_C (0x6) – PowerUpTime64Ticks_C (0x7) Note: 1 tick = 1/4096 s.
7	RW brst SPI	**PowerDownMode**: *Select Power down Mode* Selects the power down mode into which the transceiver chip is brought when the chip enters the powered down state. 0: NA_PowerDownModeFull_C Pads and transceiver registers are powered off. Reconfiguration of the transceiver registers is required after wake-up, but the leakage current is the lowest possible in this mode. 1: NA_PowerDownModePad_C All output pads are disabled and all bi-directional pads are switched to input, but all transceiver registers are still powered. Reconfiguration of the transceiver registers is not required, but leakage current is higher.

26.2.9 0x07 – Reset digital baseband/baseband clock and power down

Used for resetting the 32 MHz baseband clock distribution circuitry and the baseband radio control circuitries. It is also used for bringing the chip to configured power-down station.

Address offset: 0x07

Reset value: 0x06

RW	7	6	5	4	3	2	1	0
R						Reset BbRadio Ctrl	Reset BbClock Gate	
init	0	0	0	0	0			0
W	ST Reserved	ST Reserved						Power Down
init	0	0	0	0	0	1	1	0

Table 59. Field properties for register 0x07

Bits	Prop	Description
1	RW brst SPI	**ResetBbClockGate**: *Reset Clock Gates* Resets the 32 MHz baseband clock distribution circuitry. 0: Inactive reset 1: Active reset

Table 59. Field properties for register 0x07 (continued)

Bits	Prop	Description
2	RW brst SPI	**ResetBbRadioCtrl**: *Reset Transceiver* Resets the digital baseband and digital radio control circuitries supplied by the 32 MHz baseband clock. 0: Inactive reset 1: Active reset Note: The baseband clock must be started before the reset takes effect.
0	WO brst SPI	**PowerDown**: *Bring Chip to Powered Down State* Brings the chip to the configured power-down state. This bit will be automatically cleared when the chip is powered-up. 0: Normal operating mode 1: Go to power-down state

26.2.10 0x08 – Power on/off baseband crystal and clock

Used for selection an optional external oscillator on pad Xtal32MP, the baseband clock distribution, and the internal baseband quartz oscillator. It is also used for enabling and setting the feature clock frequency.

Address offset: 0x08

Reset value: 0x00

RW	7	6	5	4	3	2	1	0
RW	Enable Feature Clock	FeatureClockFreq			Bypass Bb Crystal		Enable BbClock	Enable BbCrystal

Table 60. Field properties for register 0x08

Bits	Prop	Description
0	RW SPI	**EnableBbCrystal**: *Enable Baseband Crystal* Powers on the internal baseband quartz oscillator. 0: Power off 1: Power on Note: Never switch on the internal oscillator when the external oscillator is connected to pad Xtal32MP.
1	RW SPI	**EnableBbClock**: *Enable Baseband Clock* Enables the baseband clock distribution. 0: Distribution off 1: Distribution on
3	RW SPI	**BypassBbCrystal**: *Bypass Baseband Crystal* Switches off the internal Oscillator when an external oscillator is connected to pad Xtal32MP. 0: Disable 1: Enable

Table 60. Field properties for register 0x08 (continued)

Bits	Prop	Description
4- 6	RW brst SPI	**FeatureClockFreq**: *Feature Clock Frequency* Sets the feature clock frequency. The feature clock can be routed to one or more digital IO pads. The frequency of the Feature Clock is calculated by using the 32/2 MHz baseband clock divided by the setting made here. – NA_FeatureClockDiv1_C (16 MHz) (0x7) – NA_FeatureClockDiv2_C (8 MHz) (0x6) – NA_FeatureClockDiv4_C (4 MHz) (0x5) – NA_FeatureClockDiv8_C (2 MHz) (0x4) – NA_FeatureClockDiv16_C (1 MHz) (0x3) – NA_FeatureClockDiv32_C (500 kHz) (0x2) – NA_FeatureClockDiv64_C (250 kHz) (0x1) – NA_FeatureClockDiv128_C (125 kHz) (0x0) Note: Change the frequency only when the feature clock is disabled (EnableFeatureClock = 0).
7	RW brst SPI	**EnableFeatureClock**: *Enable Feature Clock* Enables the Feature Clock which guarantees clip and glitch free clock waveforms on a digital IO output. 0: Disable If the Feature Clock is disabled but a digital IO pin is enabled as clock output, the frequency is equal to the real-time clock crystal frequency of 32.768 kHz. 1: Enable If enabled, the frequency depends on the value set in FeatureClockFreq (see above).

26.2.11 0x09 – Setting SPI pads as pull-up or pull-down

Used for connecting pull-up and pull-down resistors to the four SPI pads: Spiclk, Spissn, Spirxd, and Spitxd. The values programmed here are applied to the pad when they are written to the corresponding registers through register 0x0B (see *Section 26.2.13: 0x0B – Writing pull-up/pull-down settings to pads*). The resistors are kept during power-down state.

Address offset: 0x09
Reset value: 0x00

RW	7	6	5	4	3	2	1	0
R								
W	UsePulldown 4Spitxd	UsePullup 4Spitxd	UsePulldown 4Spirxd	UsePullup 4Spirxd	UsePulldown 4Spissn	UsePullup 4Spissn	UsePulldown 4Spiclk	UsePullup 4Spiclk

Table 61. Field properties for register 0x09

Bits	Prop	Description
0	WO brst SPI	**UsePullup4Spiclk**: *Connect a Pull-Up Resistor to SPI Clock Pad* When this bit is set to 1, a pull-up resistor is connected to the SPI clock pad. 0: No pull-up resistor 1: Use pull-up resistor

Table 61. Field properties for register 0x09 (continued)

Bits	Prop	Description
1	WO brst SPI	**UsePulldown4Spiclk**: *Connect a Pull-Down Resistor to SPI Clock Pad* When this bit is set to 1, a pull-down resistor is connected to the SPI clock pad, but only when UsePullup4Spiclk is 0. 0: No pull-down resistor 1: Use pull-down resistor
2	WO brst SPI	**UsePullup4Spissn**: *Connect a Pull-Up Resistor to SPI Ssn Pad* When this bit is set to 1, a pull-up resistor is connected to the SPI Ssn pad. 0: No pull-up resistor 1: Use pull-up resistor
3	WO brst SPI	**UsePulldown4Spissn**: *Connect a Pull-Down Resistor to SPI Ssn Pad* When this bit is set to 1, a pull-down resistor is connected to the SPI Ssn pad, but only when UsePullup4Spissn is 0. 0: No pull-down resistor 1: Use pull-down resistor
4	WO brst SPI	**UsePullup4Spirxd**: *Connect a Pull-Up Resistor to SPI RxD Pad* When this bit is set to 1, a resistor is connected to the SPI RxD Pad. 0: No pull-up resistor 1: Use pull-up resistor
5	WO brst SPI	**UsePulldown4Spirxd**: *Connect a Pull-Down Resistor to SPI RxD Pad* When this bit is set to 1, a pull-down resistor is connected to the SPI RxD pad, but only when UsePullup4Spirxd is 0. 0: No pull-down resistor 1: Use pull-down resistor
6	WO brst SPI	**UsePullup4Spitxd**: *Connect a Pull-Up Resistor to SPI TxD Pad* When this bit is set to 1, a resistor is connected to the SPI TxD Pad. 0: No pull-up resistor 1: Use pull-up resistor
7	WO brst SPI	**UsePulldown4Spitxd**: *Connect a Pull-Down Resistor to SPI TxD Pad* When this bit is set to 1, a pull-down resistor is connected to the SPI TxD pad, but only when UsePullup4Spitxd is 0. 0: No pull-down resistor 1: Use pull-down resistor

26.2.12 0x0A – Setting additional pads as pull-up or pull-down

Used for connecting the pull-up or the pull-down resistors to the four pads: /POnReset, Pamp, µCIrq, and µCReset. The values programmed here are applied to the pad when they are written to the corresponding registers through register 0x0B (see *Section 26.2.13: 0x0B – Writing pull-up/pull-down settings to pads*). The resistors are kept during power-down state.

Address offset: 0x0A
Reset value: 0x00

RW	7	6	5	4	3	2	1	0
R								

RW	7	6	5	4	3	2	1	0
W	Use Pulldown 4Ucrst	Use Pullup 4Ucrst	Use Pulldown 4Ucirq	Use Pullup 4Ucirq	Use Pulldown 4Pamp	Use Pullup 4Pamp	Use Pulldown 4Por	Use Pullup 4Por

Table 62. Field properties for register 0x0A

Bits	Prop	Description
0	WO brst SPI	**UsePullup4Por**: *Connect a Pull-Up Resistor to POR Pad* When this bit is set to 1, a pull-up resistor is connected to the POR pad. 0: No pull-up resistor 1: Use pull-up resistor
1	WO brst SPI	**UsePulldown4Por**: *Connect a Pull-Down Resistor to POR Pad* When this bit is set to 1, a pull-down resistor is connected to the POR pad, but only when UsePullup4Por is 0. 0: No pull-down resistor 1: Use pull-down resistor
2	WO brst SPI	**UsePullup4Pamp**: *Connect a Pull-Up Resistor to PAMP Pad* When this bit is set to 1, a pull-up resistor is connected to the PAMP pad. 0: No pull-up resistor 1: Use pull-up resistor
3	WO brst SPI	**UsePulldown4Pamp**: *Connect a Pull-Down Resistor to PAMP Pad* When this bit is set to 1, a pull-down resistor is connected to the PAMP pad, but only when UsePullup4Pamp is 0. 0: No pull-down resistor 1: Use pull-down resistor
4	WO brst SPI	**UsePullup4Ucirq**: *Connect a Pull-Up Resistor to ucIRQ Pad* When this bit is set to 1, a pull-up resistor is connected to the ucIRQ pad. 0: No pull-up resistor 1: Use pull-up resistor
5	WO brst SPI	**UsePulldown4Ucirq**: *Connect a Pull-Down Resistor to ucIRQ Pad* When this bit is set to 1, a pull-down resistor is connected to the ucIRQ pad, but only when UsePullup4Ucirq is 0. 0: No pull-down resistor 1: Use pull-down resistor
6	WO brst SPI	**UsePullup4Ucrst**: *Connect a Pull-Up Resistor to ucRST Pad* When this bit is set to 1, a pull-up resistor is connected to the ucRST pad. 0: No pull-up resistor 1: Use pull-up resistor
7	WO brst SPI	**UsePulldown4Ucrst**: *Connect a Pull-Down Resistor to ucRST Pad* When this bit is set to 1, a pull-up resistor is connected to the ucRST pad, but only when UsePullup4Ucrst is 0. 0: No pull-down resistor 1: Use pull-down resistor

26.2.13 0x0B – Writing pull-up/pull-down settings to pads

Used for writing the settings made in registers 0x09 and/or 0x0A to the appropriate pads on the chip. Because the destination registers are running with the 32.768 kHz clock, the write strobe has to be active for at least 30.6 ms.

Note: A write strobe must always be generated by programming a 1-0-sequence to this register. Do NOT change the content of registers 0x09 (see Section 26.2.11: 0x09 – Setting SPI pads as pull-up or pull-down) and 0x0A (see Section 26.2.12: 0x0A – Setting additional pads as pull-up or pull-down) when one of the bits of this register is active.

Address offset: 0x0B
Reset value: 0x00

RW	7	6	5	4	3	2	1	0
R								
W							Write Pulls4Pads	Write Pulls4Spi

Table 63. Field properties for register 0x0B

Bits	Prop	Description
0	WO brst SPI	**WritePulls4Spi**: *Write Pull-up and Pull-Down Settings to Four SPI Pads* Writes the SPI pad settings (see register 0x9) to the appropriate chip pads. 0: True 1: False
1	WO brst SPI	**WritePulls4Pads**: *Write Pull-up and Pull-Down Settings to Four Additional Pads* Writes the additional pad settings (see register 0xA) to the appropriate chip pads. 0: True 1: False

26.2.14 0x0E – Baseband memory access

Used together to selecting pages of the baseband RAM, the correlator memory, or the chirp sequencer RAM, which are then mapped into the Device Window.

Note: See also Section 6.1: Selecting a memory address on page 24.

Address offset: 0x0E
Reset value: 0x00

RW	7	6	5	4	3	2	1	0
RW			DeviceSelect				RamIndex	

Table 64. Field properties for register 0x0E

Bits	Prop.	Description
0-1	RW brst SPI	**RamIndex**: *Select Page in Baseband Memory* Selects one page of selected memory type (see DeviceSelect below) to be mapped into accessible SPI address space. When addresses are greater than 0xFF, the higher bits have to be written to this field. To select a baseband RAM page, select one of: – Baseband RAM block 0 (0x0) – Baseband RAM block 1 (0x1) – Baseband RAM block 2 (0x2) – Baseband RAM block 3 (0x3) To select a chirp sequencer (CSQ) RAM column, select one of: – Column 0 (0x0) – Column 1 (0x1) – Column 2 (0x2) – Unused (0x3) For correlator memory, select one of: – I (RamD3IPatI) (0x0) – Q (RamD3IPatQ) (0x1) – Thresholds (RamD3IThresholds) (0x2) – Unused (0x3) Note: Register access to locations < 0x80 are not influenced by index settings.
4-5	RW brst SPI	**DeviceSelect**: *Select Memory Type* Selects either the baseband RAM, the chirp sequencer RAM or the correlator memory to be mapped into the accessible SPI address space For baseband RAM, select: – NA_DeviceSelectBbRam0_C (0x0) – Unused (0x1) For chirp sequencer RAM, select: – NA_DeviceSelectCsq_C (0x2) For correlator memory, select: – NA_DeviceSelectD3I_C (0x3) Note: Register access to memory locations less than 0x80 are not influenced by index settings.

26.2.15 0x0F – LO, BBTimer, RX/TX IRQ event status and enabling

Used for selecting sources (LO, baseband timer, receiver, transmitter) that drive an external interrupt when an event occurs. It is also used for reporting the occurance of an interrupt event for one of the enabled receive/transmit interrupts.

Note: The status bits in 4 to 7 are independent of the enable bits 0 to 3.

Address offset: 0x0F
Reset value: 0x00

RW	7	6	5	4	3	2	1	0
R	LoIrq Status	BbTimer IrqStatus	RxIrq Status	TxIrq Status	LoIrq Enable	BbTimer Irq Enable	RxIrq Enable	TxIrq Enable
W								

Table 65. Field properties for register 0x0F

Bits	Prop	Description
0	RW brst SPI	**TxIrqEnable**: *Enable Transmitter Interrupt* 0: Disable - TX interrupt does not drive the interrupt line. 1: Enable - TX interrupt drives the interrupt line.
1	RW brst SPI	**RxIrqEnable**: *Enable Receiver Interrupt* 0: Disable - RX interrupt does not drive the interrupt line. 1: Enable - RX interrupt drives the interrupt line.
2	RW brst SPI	**BbTimerIrqEnable**: *Enable Baseband Timer Interrupt* 0: Disable - BbTimer interrupt does not drive the interrupt line. 1: Enable - BbTimer interrupt drives the interrupt line.
3	RW brst SPI	**LoIrqEnable**: *Enable LO Interrupt* 0: Disable - LO interrupt does not drive the interrupt line. 1: Enable - LO interrupt drives the interrupt line.
4	RO	**TxIrqStatus**: *Transmitter Interrupt Event Status* Reports the occurance of one or more of the enabled transmitter interrupts. 0: Event did not trigger This bit is cleared when an enabled (via register 0x13) transmitter interrupt is either cleared via register 0x10 or disabled via register 0x13. 1: Event is triggered This bit is 1 when one or more of the enabled (via register 0x13) transmitter interrupts have occurred. Their raw status is reported in register 0x10.
5	RO	**RxIrqStatus**: *Receiver Interrupt Event Status* Reports the occurance of one or more of the enabled receiver interrupts. 0: Event did not trigger This bit is cleared when an enabled (via register 0x14) receiver interrupt is either cleared via register 0x11 or disabled via register 0x14. 1: Event is triggered This bit is 1 when one or more of the enabled (via register 0x14) receiver interrupts have occurred. Their raw status is reported in register 0x11.
6	RO	**BbTimerIrqStatus**: *Baseband Timer Event Status* Reports the occurance of a baseband timer event. 0: Event did not trigger This bit is cleared when the BBTimer interrupt is reset via the register 0x12. 1: Event is triggered This bit is 1 when the baseband timer has reached its end.
7	RO	**LoIrqStatus**: *LO IRQ Event Status* Reports the occurance of a LO IRQ event. 0: Event did not trigger This bit is cleared when an enabled (via register 0x15) LO interrupt is either cleared via register 0x12 (WARNING: one of the sources is state triggered) or disabled via register 0x15. 1: Event is triggered This bit is 1 when one or more of the enabled (via register 0x15) LO interrupts haVE occurred. Their raw status is reported in register 0x12.

26.2.16 0x10 – TX raw interrupt status and reset [Index]

Used for reporting the TX raw interrupt status of the transmitter and for clearing the interrupt status of the corresponding source.

Address offset: 0x10
Reset value: 0x00

RW	7	6	5	4	3	2	1	0
R			TxIntsRawStat					
W			TxIntsReset					

Table 66. Field properties for register 0x10

Bits	Prop	Description
0-5	RO brst BbClk	**TxIntsRawStat**: *Transmitter Interrupt Raw Status* Reports the TX raw interrupt of each source. Values: See Bit Offsets (Indexes) section below.
0-5	WO brst strb BbClk	**TxIntsReset**: *Transmitter Interrupt Reset* Clears the interrupt status of the corresponding source. Values: See Bit Offsets (Indexes) section below.

Bit Offsets (Indexes) for TxIntsRawStat and TxIntsReset

Address offset: Index
Reset value: 0x00

7	6	5	4	3	2	1	0
		TxTime SlotEnd	TxTime SlotTOut	Tx Under run	TxEnd	TxBufferRdy	

Table 67. Field properties for TxIntsRawStat and TxIntsReset indexes

Bits	Description
0-1	**TxBufferRdy**: *Transmit Buffer Ready* Indicates that the appropriate buffer has been transmitted and can, therefore, be filled with raw data, if necessary. The LSB (bit 0x00) is related to the lower half of the TX buffer (TxRx buffer when using a combined buffer – TxRxBbBufferMode1), while the MSB (bit 0x01) is related to the upper half of the TX Buffer.
2	**TxEnd**: *Transmit End* Indicates that the transmission has ended. Conditions for successful transmissions are as follows: If ArqScheme is used for transmission (TxArq = 1) and TxArqCnt ≤ TxArqMax, then the transmission was successful. If ArqScheme is used for transmission (TxArq = 1) and TxArqCnt > TxArqMax, then the transmission was not successful. If ArqScheme is not used for transmission (TxArq =0), then transmission was always successful.

Table 67. Field properties for TxIntsRawStat and TxIntsReset indexes (continued)

Bits	Description
3	**TxUnderrun**: *Transmit Buffer Underrun* Indicates that the buffer was not filled when the transmitter tried to send data. This interrupt is intended for debugging purposes only and indicates a severe error condition of the controller. It must not occur during normal operation. The user is responsible for avoiding an underrun condition. How the transmitter has to be handled depends on the setting of the TxUnderrunIgnore field in register 0x4D.
4	**TxTimeSlotTOut**: *Transmit Time Slot Time-out* Indicates that the transmit process exceeded the time slot and that the transmission of packets was aborted automatically. Interrupt is defined only if TxTimeSlotControl = 1.
5	**TxTimeSlotEnd**: *Transmit Time Slot End* Indicates that the transmit time slot has ended. Interrupt is defined only if TxTimeSlotControl = 1.

26.2.17 0x11 – RX Raw Interrupt Status and Reset [Index]

Used for reporting the RX raw interrupt status of the receiver and for clearing the interrupt status of the corresponding source.

Address offset: 0x11
Reset value: 0x00

RW	7	6	5	4	3	2	1	0
R		RxIntsRawStat						
W		RxIntsReset						

Table 68. Field properties for register 0x1

Bits	Prop	Description
0-6	RO brst BbClk	**RxIntsRawStat**: *Receiver Interrupt Raw Status* Reports the RX raw interrupt of each source. Values: See Bit Offsets (Indexes) section below.
0-6	WO brst strb BbClk	**RxIntsReset**: *Reset Receiver Interrupt* Clears the interrupt status of the corresponding source. Values: See Bit Offsets (Indexes) section below.

Bit Offsets (Indexes) for RxIntsRawStat and RxIntsReset

Address offset: Index
Reset value: 0x00

7	6	5	4	3	2	1	0
	RxTime SlotEnd	RxTime SlotTOut	RxOverflow	Rx Header End	RxEnd	RxBufferRdy	

Bits	Description
1-0	**RxBufferRdy**: *Receive Buffer Ready* Indicates that the appropriate receive buffer is filled with data and can be read out. The LSB (0x00) is related to the lower half of the RX buffer (TxRx buffer when using a combined buffer – TxRxBbBufferMode1), while the MSB (0x01) is related to the upper half of the RX buffer.
2	**RxEnd**: *End of Packet Reception* Indicates that end of a correct packet was detected. This interrupt depends on the settings of RxCrc2Mode field and RxArqMode field in register 0x5A.
3	**RxHeaderEnd**: *End of Correct Header Reception* Indicates the reception of a correct header. Only usable in auto mode.
4	**RxOverflow**: *Receive Buffer Overflow* Indicates that the receiver was not able to place data in the RX buffer because the RX buffer was full. No RxEnd interrupt is generated and no Ack packet will be transmitted This interrupt is intended for debugging purposes only and indicates an error condition of the controller. It should not occur during normal operation. The user is responsible to avoid the overflow condition. This interrupt must be ignored when RxCrc2Mode field in register 0x5A is set to 1. Before a reception can be restarted through the RxCmdStart command, the pending RxBufferRdy interrupts and the receive buffers (through RxBufferCmd) must be cleared.
5	**RxTimeSlotTOut**: *Receive Time Slot Time-out* Indicates that the receive process exceeded the time slot and that the transmission of Clr2s or Ack packets was aborted automatically. Interrupt is defined only if RxTimeSlotControl = 1.
6	**RxTimeSlotEnd**: *Receive Time Slot End* Indicates that the receive time slot has ended. Interrupt is defined only if RxTimeSlotControl = 1.

26.2.18 0x12 – LO and BBTimer raw interrupt status and reset [Index]

Used for storing the LO interrupt of a triggered event and for clearing the interrupt status of the corresponding source. It is also used for resetting the baseband timer interrupt or for stopping the baseband timer.

Address offset: 0x12
Reset value: 0x00

RW	7	6	5	4	3	2	1	0
R							LoInts RawStat	Internal Use Only
W	Clear Base band Timer Int						LoInts Reset	Internal Use Only

Table 69. Field properties for register 0x12

Bits	Prop	Description
1	RO brst BbClk	**LoIntsRawStat**: *Local Oscillator Interrupt Raw Status* Stores the LO interrupt of a triggered event. Values: See Bit Offsets (Indexes) section below.
1	WO brst strb BbClk	**LoIntsReset**: *Reset Local Oscillator interrupt* Clears the interrupt status of the corresponding source. Values: See Bit Offsets (Indexes) section below.
7	WO brst strb BbClk	**ClearBasebandTimerInt**: *Baseband timer interrupt reset* Clears the BbTimer interrupt or stops the timer, if it has not reached zero. After ClearBasebandTimerInt, the timer can be restarted again.

Bit Offsets (Indexes) for LO Interrupts

Address offset: Index
Reset value: 0x00

7	6	5	4	3	2	1	0
						LoTuning Ready	Internal Use Only

Table 70. Field properties for LO interrupt indexes

Bits	Description
1	**LoTuningReady**: *Local Oscillator Tuning Ready* Indicates that the tuning algorithm for the LO has finished.

26.2.19 0x13 – Enable transmitter interrupts [Index]

Used for enabling the different transmitter interrupt sources to control TxIrqStatus in register 0x0F (see *Section 26.2.15: 0x0F – LO, BBTimer, RX/TX IRQ event status and enabling*).

Address offset: 0x13
Reset value: 0x00

RW	7	6	5	4	3	2	1	0
R								
W			TxIntsEn					

Table 71. Field properties for register 0x13

Bits	Prop	Description
0-5	WO brst BbClk	**TxIntsEn**: *Enable Transmitter Interrupt* Enables the different TX interrupt sources to control TxIrqStatus. Values: See Bit Offsets (Indexes) section below.

Bit Offsets (Indexes) for TxIntsEn

Address offset: Index
Reset value: 0x00

7	6	5	4	3	2	1	0
		TxTime SlotEnd	TxTime SlotTOut	Tx Underrun	TxEnd	TxBufferRdy	

Table 72. Field properties for TxIntsEn indexes

Bits	Description
0-1	**TxBufferRdy**: *Transmit Buffer Ready* Indicates that the appropriate buffer has been transmitted and can be filled with data. The LSB (bit 0x00) is related to the lower half of the TX buffer (TxRx buffer when using a combined buffer – TxRxBbBufferMode1), while the MSB (bit 0x01) is related to the upper half of the TX buffer.
2	**TxEnd**: *Transmit End* Indicates that the transmission has ended. Conditions for a successful transmission are as follows: – If ArqScheme is used for transmission (TxArq = 1) and TxArqCnt ≤ TxArqMax, then the transmission was successful. – If ArqScheme is used for transmission (TxArq = 1) and TxArqCnt > TxArqMax, then the transmission was not successful. – If ArqScheme is not used for transmission (TxArq =0) then transmission was always successful.
3	**TxUnderrun**: *Transmit Buffer Underrun* Indicates that buffer was not filled when the transmitter tried to send data. This interrupt is intended for debugging purposes only and indicates a severe error condition of the controller. It must not occur during normal operation. The user is responsible for avoiding the underrun condition. How the transmitter has to be handled depends on the setting of the TxUnderrunIgnore field in register 0x4D.
4	**TxTimeSlotTOut**: Transmit Time Slot Time-out Indicates that the transmit process exceeded the time slot and that the transmission of packets was aborted automatically. Interrupt is defined only if TxTimeSlotControl = 1.
5	**TxTimeSlotEnd**: *Transmit Time Slot End* Indicates that the transmit time slot has ended. Interrupt is defined only if TxTimeSlotControl = 1.

26.2.20 0x14 – Receiver interrupt enable [Index]

Used for enabling the different receiver interrupt sources to control RxIrqStatus in register 0x0F (see *Section 26.2.15: 0x0F – LO, BBTimer, RX/TX IRQ event status and enabling*).

Address offset: 0x14
Reset value: 0x00

RW	7	6	5	4	3	2	1	0
R								
W		RxIntsEn						

Table 73. Field properties for register 0x14

Bits	Prop	Description
0-6	WO brst BbClk	**RxIntsEn**: *Enable Receiver Interrupt* Enables the different RX interrupt sources to control the combined RxIrqStatus. Values: See Bit Offsets (Indexes) section below.

Bit offsets (Indexes) for RX interrupts

Address offset: Index
Reset value: 0x00

7	6	5	4	3	2	1	0
	RxTime SlotEnd	RxTime SlotTOut	RxOver flow	RxHeader End	RxEnd	RxBufferRdy	

Table 74. Field properties for RX interrupt indexes

Bits	Description
1-0	**RxBufferRdy**: *Receive Buffer Ready* Indicates that the appropriate receive buffer is filled with data and can be read out. The LSB (0x00) is related to the lower half of the RX buffer (TxRx buffer when using a combined buffer – TxRxBbBufferMode1), while the MSB (0x01) is related to the upper half of the RX buffer.
2	**RxEnd**: *End of Packet Reception* Indicates that end of a correct packet was detected. This interrupt depends on the settings of RxCrc2Mode field and RxArqMode field in register 0x5A.
3	**RxHeaderEnd**: *End of Correct Header Reception* Indicates the reception of a correct header. Only usable in auto mode.
4	**RxOverflow**: *Receive Buffer Overflow* Indicates that the receiver was not able to place data in the receive buffer because the receive buffer was full. No RxEnd interrupt is generated and no Ack packet will be transmitted. This interrupt is intended for debugging purposes only and indicates an error condition of the controller. It should not occur during normal operation. The user is responsible to avoid the overflow condition. This interrupt must be ignored when RxCrc2Mode field in register 0x5A is set to 1. Before a reception can be restarted through the RxCmdStart command, the pending RxBufferRdy interrupts and the receive buffers (through RxBufferCmd) must be cleared.
5	**RxTimeSlotTOut**: *Receive Time Slot Time-out* Indicates that the receive process exceeded the time slot and that the transmission of Clr2s or Ack packets was aborted automatically. Interrupt is defined only if RxTimeSlotControl = 1.
6	**RxTimeSlotEnd**: *Receive Time Slot End* Indicates that the receive time slot has ended. Interrupt is defined only if RxTimeSlotControl = 1.

26.2.21 0x15 – Local oscillator interrupt enable [index]

Used for enabling the different LO interrupt sources to control the combined LoIrqStatus in register 0x0F (see *Section 26.2.15: 0x0F – LO, BBTimer, RX/TX IRQ event status and enabling*).

Address offset: 0x15
Reset value: 0x00

RW	7	6	5	4	3	2	1	0
R								
W							LoIntsEn	Internal Use Only

Table 75. Field properties for register 0x15

Bits	Prop	Description
1	WO	**LoIntsEn**: *Enable Local Oscillator interrupt* Enables the different LO interrupt sources to control the combined LoIrqStatus. Values: See Indexes section below.

Bit offsets (indexes) for LO interrupts

Address offset: Index
Reset value: 0x00

7	6	5	4	3	2	1	0
						LoTuning Ready	Internal Use Only

Table 76. Field properties for LO interrupts

Bits	Description
1	**LoTuningReady**: *Local Oscillator Tuning Ready* Indicates that the tuning algorithm for the LO has finished.

26.2.22 0x16 to 0x18 – RF local oscillator RX capacitors

Used for reading the 22 switchable capacitors of the Local Oscillator after tuning as well as for setting the 22 switchable capacitors of the Local Oscillator for the RX frequencies.

Note: To save time, the capacitor values for receive can be read after the adjustment has been performed. After following a power-down and wake-up, these values can be written to the register instead of adjusting the Local Oscillator again.

Offset	RW	7	6	5	4	3	2	1	0
0x16	RW	LoRxCapsValue							
	init	0	1	0	0	0	0	0	0
0x17	RW	LoRxCapsValue							
	init	0	0	0	0	0	0	0	0
0x18	RW			LoRxCapsValue					
	init	0	0	1	0	0	0	0	0

Table 77. Field properties for registers 0x16 to 0x18

Bits	Prop	Description
0-21	RW brst BbClk	**LoRxCapsValue**: *Read or Write the Caps Value for RX* Reads or writes the 22 capacitors of the Local Oscillator for receive frequency. The correct capacitor values can be read after the Local Oscillator has been tuned. Default: 0x200040

26.2.23 0x19 to 0x1B – RF local oscillator TX capacitors

Used for reading the 22 switchable capacitors of the Local Oscillator after tuning as well as for setting the 22 switchable capacitors of the Local Oscillator for the TX frequencies.

Note: To save time, the capacitor values for transmit can be read after the adjustment has been performed. After following a power-down and wake-up, these values can be written to the register instead of adjusting the Local Oscillator again.

Offset	RW	7	6	5	4	3	2	1	0
0x19	RW	LoTxCapsValue							
	init	0	1	0	0	0	0	0	0
0x1A	RW	LoTxCapsValue							
	init	0	0	0	0	0	0	0	0
0x1B	RW			LoTxCapsValue					
	init	0	0	1	0	0	0	0	0

Table 78. Field properties for registers 0x19 to 0x1B

Bits	Prop	Description
0-21	RW brst BbClk	**LoTxCapsValue**: Read *or Write the Caps Value for TX* Reads or writes the 22 capacitors of the Local Oscillator for transmit frequency. The correct capacitor values can be read after the Local Oscillator has been tuned. Default: 0x200040

26.2.24 RF local oscillator controls

Used for tuning the LO capacitors to the target frequency and for setting the level of accuracy of the tuning. It is also used for accelerating the LO caps tuning, for enabling auto-recalibration of the LO caps, and for switching between TX and RX caps for LO caps tuning.

Address offset: 0x1C
Reset value: 0x00

RW	7	6	5	4	3	2	1	0
R								
W	UseLo RxCaps		Internal Use Only	LoEnable LsbNeg	LoFastTuningLevel			LoEnable FastTuning

Table 79. Field properties for register 0x1C

Bits	Prop	Description
0	WO brst BbClk	**LoEnableFastTuning**: *Enable Accelerated LO Caps Tuning* Enables an algorithm that speeds up the tuning of the LO capacitors to the target frequency. 0: Normal tuning 1: Accelerated tuning
1-3	WO brst BbClk	**LoFastTuningLevel**: *Set LO Caps Tuning Level* Sets the level of accuracy of the tuning. Generally, the higher the tuning level, the faster the tuning to the target frequency. Allowed Values: – If LoEnableLsbNeg = = 0, then 0x0 to 0x1 (maximum 2) – If LoEnableLsbNeg = = 1, then 0x0 to 0x4
4	WO brst BbClk	**LoEnableLsbNeg**: *Enable Higher Precision Accelerated LO Caps Tuning* Accelerates the LO Caps tuning by factor of 2. 0: Normal tuning 1: Accelerated tuning by factor of 2
7	WO brst BbClk	**UseLoRxCaps**: *Select RX or TX Caps* Switch between TX and RX Caps for LO caps tuning. 0: Use TX caps 1: Use RX caps This field must be set before starting the LO caps tuning.

26.2.25 0x1D to 0x1E – RF local oscillator target value

Used for setting the target value for the LO center frequency, which is used for adjusting either the RX or TX Local Oscillator frequency.

Offset	RW	7	6	5	4	3	2	1	0
0x1D	RW	LoTargetValue							
	init	0	0	0	0	0	0	0	0
0x1E	RW	LoTargetValue							
	init	0	0	0	0	0	0	0	0

Table 80. Field properties for registers 0x1D to 0x1E

Bits	Prop	Description
0-16	WO brst BbClk	**LoTargetValue**: *Target Value for Center Frequency* 16-bit target value for the Local Oscillator frequency adjustment. This value is used to determines the target LO center frequency. The LOTargetValue is determined as follows: $\lceil (\{LOdiv10/(16MHz)\} - 12) \cdot 8192 \rceil$ Before setting this value, the following must be enabled: – Local Oscillator – EnableLO – Clock divider – EnableLOdiv10 Also, the appropriate mode (RX, TX) must be selected using the UseLoRxCaps: – UseLoRxCaps = 1 (RX mode) – UseLoRxCaps = 0 (Tx mode) A write operation to the upper byte starts the tuning algorithm automatically.

26.2.26 0x1F – AGC threshold 1

Used for setting the lower threshold of the integration value.

Offset	RW	7	6	5	4	3	2	1	0
0x1F	R								
	init	0	0	0	0	0	0	0	0
	W	AgcThresHold1							
	init	0	0	0	0	0	0	1	1

Table 81. Field properties for register 0x1F

Bits	Prop	Description
0-7	WO brst BbClk	**AgcThresHold1**: *Set AGC Threshold1* Sets the lower level of the integration value. Default: 0x3

26.2.27 0x20 – AGC threshold 2

Used with AgcThresHold1 for setting the upper threshold of the integration value.

Offset	RW	7	6	5	4	3	2	1	0
0x20	R								
	init	0	0	0	0	0	0	0	0
	W	AgcThresHold2							
	init	0	0	0	0	0	1	1	0

Table 82. Field properties for register 0x20

Bits	Prop	Description
0-7	WO brst BbClk	**AgcThresHold2**: *Set AGC Threshold2* AgcThresHold1 plus the value set in this register is used to determine the upper level of the integration value. Default: 0x6

26.2.28 0x21 – AGC hold frame/bit synchronization controls

Used for enabling or disabling the AGC stop and for defining a delay from the bit synchronization state until the AGC stop is enabled in bit synchronized state.

Offset	RW	7	6	5	4	3	2	1	0
0x21	R								
	init	0	0	0	0	0	0	0	0
	W	HoldAgcInFrameSync	HoldAgcInBitSync						
	init	1	0	0	1	1	0	0	0

Table 83. Field properties for register 0x21

Bits	Prop	Description
0-6	WO brst BbClk	**HoldAgcInBitSync**: *RF AGC Hold Bit Synchronization Control* When in bit synchronization state, gain is not changed (AGC is stopped) after a programmable number of pulses are received. Default: 0x18
7	WO brst BbClk	**HoldAgcInFrameSync**: *RF AGC Hold Frame Synchronization Control* When the receiver reaches frame synchronization state, gain is not changed (AGC is stopped). 0: Do not hold AGC 1: Hold AGC Default: 0x1

26.2.29 0x22 – AGC change gain length

Used for setting the AGC gain change mode.

Note: When using this register, bits 0 to 5 MUST be set to 0xC.

Offset	RW	7	6	5	4	3	2	1	0
0x22	R								
	init	0	0	0	0	0	0	0	0
	W	AgcNregLength		Internal Use Only					
	init	0	0	0	0	1	1	0	0

Table 84. Field properties for register 0x22

Bits	Prop	Description
6-7	WO brst BbClk	**AgcNregLength**: *AGC Change Gain Length* Sets the gain change mode. When the AGC has changed (N-1) times during the last N rounds in the same direction, then fast AGC tuning is restarted. 00: 00: N=5 01: 01: N=7 10: 00: N=10 Default: 0x0

26.2.30 0x23 to 0x24 – AGC integration time

Used for setting the number of cycles for AGC integration.

Note: One cycle is equal to LO frequency divided by 20.

Offset	RW	7	6	5	4	3	2	1	0
0x23	R								
	init	0	0	0	0	0	0	0	0
	W	AgcIntTime							
	init	0	0	0	0	0	0	0	0
0x24	R								
	init	0	0	0	0	0	0	0	0
	W					AgcIntTime			
	init	0	0	0	0	0	0	1	0

Table 85. Field properties for registers 0x23 to 0x24

Bits	Prop	Description
0-12	WO brst BbClk	**AgcIntTime**: *AGC Integration Time* Sets the number of cycles for AGC integration. Default: 0x200 One cycle is equal to LO frequency divided by 20.

26.2.31 0x25 – AGC value

Used for setting the AGC value.

Offset	RW	7	6	5	4	3	2	1	0
0x25	R								
	init	0	0	0	0	0	0	0	0
	W	AgcHold	AgcDefault En	AgcValue					
	init	0	0	1	1	1	1	1	1

Table 86. Field properties for register 0x25

Bits	Prop	Description
0-5	WO brst BbClk	**AgcValue**: *Set AGC Value* When AgcHold and AgcDefaultEn are set to 1 this value is used for the RX amplifiers. Recommended value: 0x23
6	WO brst BbClk	**AgcDefaultEn**: *Enable AGC Default Value* Uses AgcValue for Rx amplifiers when AGC is held. 0: Ignore AGC value. Uses actual value from AGC loop. 1: Use default AgcValue Recommended value: 0x0

Table 86. Field properties for register 0x25 (continued)

Bits	Prop	Description
7	WO brst BbClk	**Agchold**: *Enable RF AGC Hold* Hold AGC value (stop AGC). 0: AGC active (still running) 1: Hold AGC using specified AgcValue if AgcDefaultEn = 1 or use actual AGC value (from AGC loop) if AgcDefaultEn = 0. Recommended value: 0x0

26.2.32 0x26 – RF AGC controls

Used for setting the AGC gain threshold value and for reading the AGC voltage value.

Offset	RW	7	6	5	4	3	2	1	0
0x26	R			AgcGain					
	init	0	0	0	0	0	0	0	0
	W	Internal Use Only		AgcGainThres					
	init	0	0	0	1	1	1	1	0

Table 87. Field properties for register 0x26

Bits	Prop	Description
0-5	RO brst BbClk	**AgcGain**: *Reads the AGC value for Current Packet* The RX gain used for the current packet can be read after the packet header end interrupt. Auto Mode only
0-5	WO brst BbClk	**AgcGainThres**: *AGC Gain Threshold* When gain is below this threshold, media is interpreted as busy when the upper bit TxPhCarrSenseMode field is set to 1. Default: 0x1A

26.2.33 0x27 – RC oscillator tuning

Used for initiating an FCT (Filter Caps Tuning) count cycle and for switching on the FCT generator. It is also used for switching the capacitors of the baseband low-pass filter, for enabling the amplitude of the transmitter and for reading the FCT counter. See also *Section 22.1: Adjusting the baseband filter frequency on page 94*.

Offset	RW	7	6	5	4	3	2	1	0
0x27	R			FctPeriod					
	init	0	0	0	0	0	0	0	0
	W	Enable Tx		StartFct Measure	Fct ClockEn	ChirpFilterCaps			
	init	0	0	0	0	0	1	1	0

Table 88. Field properties for register 0x27

Bits	Prop	Description
0-5	RO brst BbClk	**FctPeriod**: *RF TX FCT Counter* Reports the return value from tuning. It counts the amount of 16 MHz clock periods during a FCT clock cycle.
0-3	WO brst BbClk	**ChirpFilterCaps**: *RF TX Chirp Filter Capacitors* Switches the capacitors of the chirp filter to adjust the FCT clock to 400 kHz. Default: 0x06
4	WO brst BbClk	**FctClockEn**: *Enable RF TX FCT* Switches on the FCT generator. This is required for the chirp filter calibration. 0: Disable 1: Enable Default: 0x0
5	WO brst BbClk	**StartFctMeasure**: *Set RF TX FCT Measurement* Starts the tuning algorithm. It initiates an FCT count cycle. 0: Ignore 1: Start algorithm Default: 0x0 Note: FctClockEn must be set to 1 before starting the measurement cycle.
7	WO brst BbClk	**EnableTx**: *Enable RF TX* Enables the transmitter for tuning. It manually starts the transmitter. 0: Disable manual starting of TX 1: Manual starting of TX Default: 0x0

26.2.34 0x28 to 0x29 – Baseband timer start value

Used for setting the start value of the baseband timer. The baseband timer is running with 1 MHz. It starts when the upper byte of BasebandTimerStartValue is written. The baseband timer is stopped when the timer has expired or when writing ClearBasebandTimerInt.

Offset	RW	7	6	5	4	3	2	1	0
0x28	R								
	W	BasebandTimerStartValue							
	init	0	0	0	0	0	0	0	0
0x29	R								
	W	BasebandTimerStartValue							
	init	0	0	0	0	0	0	0	0

Table 89. Field properties for registers 0x28 to 0x29

Bits	Prop	Description
0-15	WO brst BbClk	**BasebandTimerStartValue**: *Baseband Timer Start Value* Sets the start value of the timer. The interrupt BbTimerInt is triggered if the timer reaches zero. The timer starts automatically when the upper byte is written to the register. Note: Timer tick period is 1 µs. Timer values range from 1 to 65535.

26.2.35 0x2A to 0x2B – ToaOffset for received Ack packet

Used for reporting the Time of Arrival Offset of a received Ack packet, and for reporting if the time measurement is valid for the Ack packet.

Offset	RW	7	6	5	4	3	2	1	0
0x2A	R	ToaOffsetMeanAck							
	init	0	0	0	0	0	0	0	0
0x2B	R	Toa Offset Mean AckValid			ToaOffsetMeanAck				
	init	0	0	0	0	0	0	0	0

Table 90. Field properties for registers 0x2A to 0x2B

Bits	Prop	Description
0-12	RO BbClk	**ToaOffsetMeanAck**: *ToA Offset for Received Ack Packet* Reports the Time of Arrival Offset of a received Ack packet.
15 (7)	RO BbClk	**ToaOffsetMeanAckValid**: *Valid Ack Packet* Reports that enough pulses were received at end of Ack packet.

26.2.36 0x2C to 0x2D – Round trip time

Used for reporting the round trip time of a transmitted Data packet and of a received Ack packet.

Offset	RW	7	6	5	4	3	2	1	0
0x2C	R	TxRespTime							
	init	0	0	0	0	0	0	0	0
0x2D	R	TxRespTime							
	init	0	0	0	0	0	0	0	0

Table 91. Field properties for registers 0x2C to 0x2D

Bits	Prop	Description
0-15	RO BbClk	**TxRespTime**: *TX Response Round Trip Time* Reports the response time of the transmission of a Data packet and the reception of the corresponding Acknowledgement (one round trip).

26.2.37 0x2E – Phase offset for RX data and Ack packets

Used for reporting the phase offset for received Data and Ack packets.

Offset	RW	7	6	5	4	3	2	1	0
0x2E	R		PhaseOffsetData				PhaseOffsetAck		
	init	0	0	0	0	0	0	0	0

Table 92. Field properties for register 0x2E

Bits	Prop	Description
0-2	RO BbClk	**PhaseOffsetAck**: *PhaseOffset for Received Ack Packet* Reports the phase offset of a received Ack packet.
4-6	RO BbClk	**PhaseOffsetData**: *PhaseOffset for Received Data Packet* Reports the phase offset of a received Data packet.

26.2.38 0x2F to 0x30 – ToaOffset for received data packet (read only)

Used for reporting the Time of Arrival of a received Data packet, and for reporting if enough pulses were received to provide a valid time measurement.

Offset	RW	7	6	5	4	3	2	1	0
0x2F	R	ToaOffsetMeanData							
	init	0	0	0	0	0	0	0	0
0x30	R	ToaOffset MeanData Valid			Internal Use Only				
	init	0	0	0	0	0	0	0	0

Table 93. Field properties for registers 0x2F to 0x30

Bits	Prop	Description
0-12	RO BbClk	**ToaOffsetMeanData**: *ToA Offset for Received Data Packet* Reports the time of arrival of a received Data packet.
15 (7)	RO BbClk	**ToaOffsetMeanDataValid**: *Valid Data Packet* Reports if enough pulses were received to provide a valid time measurement.

26.2.39 0x2A to 0x31 – Transceiver SyncWord (write only)

Used for defining the SyncWord, which is used for frame synchronization during reception (correlation).

Offset	RW	7	6	5	4	3	2	1	0
0x2A	W	SyncWord							
	init	1	0	1	0	1	0	1	1
0x2B	W	SyncWord							
	init	0	0	1	0	1	1	0	0
0x2C	W	SyncWord							
	init	1	1	0	1	0	1	0	1
0x2D	W	SyncWord							
	init	1	0	0	1	0	0	1	0
0x2E	W	SyncWord							
	init	1	0	0	1	0	1	0	0
0x2F	W	SyncWord							
	init	1	1	0	0	1	0	1	0
0x30	W	SyncWord							
	init	0	1	1	0	1	0	0	1

Offset	RW	7	6	5	4	3	2	1	0
0x31	W	SyncWord							
	init	1	0	1	0	1	0	1	1

Table 94. Field properties for registers0x2A to 0x31

Bits	Prop	Description
0-63	WO brst BbClk	**SyncWord**: *Set Transceiver SyncWord* Defines the synchronization word which is used for frame synchronization. Default: 0xAB69CA9492D52CAB Usage: Auto and transparent mode. TxCmdStart = True or RxCmdStart = True

26.2.40 0x31 – Receiver mode controls (read only)

Used for reporting the status of CRC1 and CRC2 checks, and for reporting the stored status of the address matching circuitry and the received packet type.

Offset	RW	7	6	5	4	3	2	1	0
0x31	R	RxCrc2 Stat	RxCrc1 Stat	RxAddrMatch		RxPacketType			
	init	0	0	0	0	0	0	0	0

Table 95. Field properties for register 0x31

Bits	Prop	Description
0-3	RO brst BbClk	**RxPacketType**: *Received Packet Type* Stores the received packet type. – Data packet = 0x0 – TimeB packet = 0x2 – Broadcast packet = 0x3 Note: Valid only when the RxHeaderEnd event is triggered. Usage: Auto mode only.
4-5	RO brst BbClk	**RxAddrMatch**: *Receive Address Match for Data Packets* Reports the address matching status of Data packets. It is reported if either the destination address is equal the station address set in RamStaAddr0 or if the destination address, with ES mode disabled, is equal RamStaAddr1. When RxAddrMatch[0]: – 0 = Received destination address does not match the station address set in RamStaAddr0 – 1 = Received destination address matches the station address set in RamStaAddr0 When RxAddrMatch[1]: – 0 = Received destination address does not match the station address set in RamStaAddr1 – 1 = Received destination address matches the station address set in RamStaAddr1 Note: Address matching must be enabled (RxAddrMode). Also, valid only when the RxHeaderEnd event is triggered. Usage: Auto mode only.

Table 95. Field properties for register 0x31 (continued)

Bits	Prop	Description
6	RO brst BbClk	**RxCrc1Stat**: *Status of CRC1 Check (For MAC Header)* Reports the status of the CRC1 check. 0: CRC1 check has failed 1: CRC1 check was successful Note: Valid only when the RxHeaderEnd event is triggered. This register just makes sense for TimeB packets when RxTimeBCrc1Mode == NA_RxTimeBCrc1ModeOn_BC_C. Usage: Auto mode only.
7	RO brst BbClk	**RxCrc2Stat**: *Status of CRC2 Check (For Payload)* Reports the status of the CRC2 check. 0: CRC2 check has failed 1: CRC2 check was successful Note: Valid only when the RxEnd event is triggered. Usage: Auto mode only.

26.2.41 0x32 – Receive correlator error controls

Used for defining the maximum allowed bit errors when a received pattern is correlated to the SyncWord. It is also used for reading the number of bit errors occurred during SyncWord correlation.

Offset	RW	7	6	5	4	3	2	1	0
0x32	R					RxCorrBitErr			
	init	0	0	0	0	0	0	0	0
	W	RxCorrErrThres							
	init	0	0	1	1	0	0	0	0

Table 96. Field properties for register 0x32

Bits	Prop	Description
0-3	RO brst BbClk	**RxCorrBitErr**: *Number of Syncword Errors* Reports the number of bit errors that have occurred during Syncword correlation. Note: Register is valid when the RxHeaderEnd event is triggered. Usage: Auto mode only.
4-7	WO brst BbClk	**RxCorrErrThres**: *Maximum Allowable Syncword Errors* It defines the threshold for bit errors allowed during the correlation of a received pattern against the Syncword stored in register 0x2A (SyncWord). If the occurred bit errors are equal to or less than RxCorrErrThres, then the receiver synchronizes. Allowed values are between 0 and 15. Default: 0x3 This register must be defined before: RxCmdStart = True Usage: Auto and transparent mode.

26.2.42 0x33 to 0x34 – Transmit time slot start (write only)

Used for defining the beginning of the time slot for a transmission.

Offset	RW	7	6	5	4	3	2	1	0
0x33	W	TxTimeSlotStart							
	init	0	0	0	0	0	0	0	0
0x34	W	TxTimeSlotStart							
	init	0	0	0	0	0	0	0	0

Table 97. Field properties for registers 0x33 to 0x34

Bits	Prop	Description
0-15	WO brst BbClk	**TxTimeSlotStart**: *Set Transmit Time Slot Start* Defines the beginning of the time slot for a transmission. Used with TxTimeSlotControl. Only the 16 LSBs of the RTC value are compared to TxTimeSlotStart. If both values are equal, then the time slot begins. This register must be defined before: TxCmdStart = True. Usage: Auto mode only.

26.2.43 0x35 to 0x36 – Transmit time slot end (write only)

Used for defining the end of the time slot for a transmission.

Offset	RW	7	6	5	4	3	2	1	0
0x35	W	TxTimeSlotEnd							
	init	0	0	0	0	0	0	0	0
0x36	W	TxTimeSlotEnd							
	init	0	0	0	0	0	0	0	0

Table 98. Field properties for registers 0x35 to 0x36

Bits	Prop	Description
0-15	WO brst BbClk	**TxTimeSlotEnd**: *Set Transmit Time Slot End* Defines the end of the time slot for a transmission. Used with TxTimeSlotControl. Only the 16 LSBs of the RTC value are compared to TxTimeSlotEnd. If both values are equal, then the time slot ends. This register must be defined before: TxCmdStart = True Usage: Auto mode only.

26.2.44 0x33 – Receiver encryption (read only)

Used for reporting the status of End Station and Intermediate Station address matching, as well as for reporting the value of the three encryption control fields: CryptEn, CryptID, and CryptSeqN.

Offset	RW	7	6	5	4	3	2	1	0
0x33	R	RxCrypt SeqN	RxCryptId		RxCrypt En			RxAddr Seg IsMatch	RxAddr Seg EsMatch
	init	0	0	0	0	0	0	0	0

Table 99. Field properties for register 0x33

Bits	Prop	Description
0	RO brst BbClk	**RxAddrSegEsMatch**: *RX Address Segment End Station (ES) Match* Shows the status of the segmented address match of end station address part. 0: End Station address does not match 1: End Station address matches
1	RO brst BbClk	**RxAddrSegIsMatch**: *RX Address Segment Intermediate Station (IS) Match* Shows the status of the segmented address match of Intermediate Station address part. 0: Intermediate Station address does not match 1: Intermediate Station address matches
4	RO brst BbClk	**RxCryptEn**: *Value of CryptEn Bit* Reads the value of the one bit Encryption Enable field, which is in the four bit EncryptionControl field inside the MAC header. 0: Disabled 1: Enabled Note: Valid when the RxHeaderEnd event is triggered.
5-6	RO brst BbClk	**RxCryptId**: *Value of CryptID Bits* Reads the value of the two bit Encryption ID field for encrypted Data, Brdcast, and TimeB packets, which is in the four bit EncryptionControl field inside the MAC header. Note: Valid when the RxHeaderEnd event is triggered.
7	RO brst BbClk	**RxCryptSeqN**: *Value of CryptSeqN Bit* Reads the value of the one bit sequential numbering scheme for encryption. The receiver compares the last received against the present CryptSeqN bit and increments its crypt clock value by one, if they are not identical. Note: Valid when the RxHeaderEnd event is triggered.

26.2.45 0x34 to 0x35 – Receive FEC single bit error count (read only)

Used for reporting the number of single bit errors that have occurred in a received packet.

Offset	RW	7	6	5	4	3	2	1	0
0x34	R	RxFec1BitErr							
	init	0	0	0	0	0	0	0	0
0x35	R		RxFec1BitErr						
	init	0	0	0	0	0	0	0	0

Table 100. Field properties for registers 0x34 to 0x35

Bits	Prop	Description
0-14	RO brst BbClk	**RxFec1BitErr**: *Number of Corrected FEC Errors* Reports the number of correctable single bit errors that have occurred in a receive packet. Note: Register is valid when the event RxEnd is triggered and when UseFec == UseFecOn. Usage: Auto mode only.

26.2.46 0x37 – Transmitter and receiver time slot control (TDMA)

Used for enabling TX time slot control (TDMA) and RX time slot control (TDMA).

Offset	RW	7	6	5	4	3	2	1	0
0x37	R								
	init	0	0	0	0	0	0	0	0
	W							RxTime SlotControl	TxTime SlotControl
	init	0	0	0	0	0	0	0	0

Table 101. Field properties for register 0x37

Bits	Prop	Description
0	WO brst BbClk	**TxTimeSlotControl**: *Enable TX Time Slot Control (TDMA)* Enables the TX time slot control so that packets are transmitted only during the defined TX time slot. 0: Disabled. Packets are transmitted independent of the time slot. 1: Enabled. Packets are transmitted only during the defined TX time slot. This register must be defined before: TxCmdStart = True Usage: Auto mode only.
1	WO brst BbClk	**RxTimeSlotControl**: *Receive Time Slot Control* Enables the RX time slot control so that at the beginning of the defined RX time slot, the receiver is automatically activated. 0: Disabled. Activates the receiver independent of the defined time slot through the RxCmd command. 1: Enabled. Automatically activates the receiver at the beginning of the defined RX time slot. Note: When enabled, the receive can still be manually activated through the RxCmd command. Usage: Auto mode only.

26.2.47 0x38 to 0x39 – RX packet slot (read only)

Used for reporting the start time slot for receiving a packet by reading the 16 LSBs of the RTC value on a header end event.

Note: Bit 0 of 0x38 is LSB and Bit 7 of 0x39 is MSB.

Offset	RW	7	6	5	4	3	2	1	0
0x38	R	RxPacketSlot							
	init	0	0	0	0	0	0	0	0
0x39	R	RxPacketSlot							
	init	0	0	0	0	0	0	0	0

Table 102. Field properties for registers 0x38 to 0x39

Bits	Prop	Description
0-15	RO brst BbClk	**RxPacketSlot**: *Time (RTC) at End of Header* Reports time (16 LSBs of RTC) at the packet header end event. Register is valid when the RxHeaderEnd event is triggered. Usage: Auto mode only.

26.2.48 0x38 to 0x39 – RX time slot start and packet slot (write only)

Used for defining the start time slot for a reception as well as for reading the 16 LSBs of the RTC value on a header end event.

Note: Bit 0 of 0x38 is LSB and Bit 7 of 0x39 is MSB.

Offset	RW	7	6	5	4	3	2	1	0
0x38	W	RxTimeSlotStart							
	init	0	0	0	0	0	0	0	0
0x39	W	RxTimeSlotStart							
	init	0	0	0	0	0	0	0	0

Table 103. Field properties for registers 0x38 and 0x39

Bits	Prop	Description
0-15	WO brst BbClk	**RxTimeSlotStart**: *Set Receive Time Slot Start* Defines the beginning of the time slot for a reception. Used with RxTimeSlotControl. Only the 16 LSBs of the RTC value are compared to RxTimeSlotStart. If both values are equal, the time slot begins. This register must be defined before: RxCmdStart = True Usage: Auto mode only.

26.2.49 0x3A to 0x3B – RX time slot end

Used for defining the end of the time slot for reception.

Offset	RW	7	6	5	4	3	2	1	0
0x3A	R								
	init	0	0	0	0	0	0	0	0
	W	RxTimeSlotEnd							
	init	0	0	0	0	0	0	0	0
0x3B	R								
	init	0	0	0	0	0	0	0	0
	W	RxTimeSlotEnd							
	init	0	0	0	0	0	0	0	0

Table 104. Field properties for registers 0x3A to 0x3B

Bits	Prop	Description
0-15	WO brst BbClk	**RxTimeSlotEnd**: *Set Receive Time Slot End* Defines the end of the time slot for a reception. Used with RxTimeSlotControl. Only the 16 LSBs of the RTC value are compared to RxTimeSlotEnd. If both values are equal, the time slot ends. This register must be defined before: RxCmdStart = True Usage: Auto mode only.

26.2.50 0x3C – ARQ

Used for setting the maximum value for packet retransmissions and for reporting the ARQ retransmission counter value.

Offset	RW	7	6	5	4	3	2	1	0
0x3C	R					TxArqCnt			
	init	0	0	0	0	0	0	0	0
	W	TxArqMax							
	init	1	1	1	0	0	0	0	0

Table 105. Field properties for register 0x3C

Bits	Prop	Description
0-3	RO brst BbClk	**TxArqCnt**: *Transmitter ARQ Count (Required Retransmissions)* Reports the required number of retransmissions. The following conditions are used to determine if the transmission was successful or unsuccessful: – If TxArqCnt <= TxArqMax, then the transmission was successful. – If TxArqCnt > TxArqMax, then the transmission was not successful.
4-7	WO brst BbClk	**TxArqMax**: *Transmit ARQ Maximum* Sets the maximal value for packet retransmissions. The maximal allowed value is 14. This register must be defined before: TxCmdStart = True Usage: Auto mode only.

26.2.51 0x3D – RF chirp generator controls

Used for providing settings for the chirp generator and for accessing the Chirp Sequencer RAM.

Offset	RW	7	6	5	4	3	2	1	0
0x3D	R								
	init	0	0	0	0	0	0	0	0
	W	Internal Use Only	Csq UseRam	Csq Mem AddrInit	Csq Asy Mode	Csq Use 4Phases	CsqUse Phase Shift	CsqDitherValue	
	init	0	0	0	0	0	1	0	0

Table 106. Field properties for register 0x3D

Bits	Prop	Description
0-1	WO brst BbClk	**CsqDitherValue**: *Chirp Sequencer Dither Value* Provides a dither function, which is used to delay transmission based on settings made to this field. Each cycle is equal to the frequency of LO div 20. 0 = No delay 1 = delay is 0 or 1 cycle (distribution 1:1) 2 = delay is 0, 1, or 2 cycles (distribution 1:1:1) 3 = delay is 0, 1, or 2 cycles (distribution 2:1:1)
2	WO brst BbClk	**CsqUsePhaseShift**: *Chirp Sequencer Use Phase Shift* Enables the pseudo-random feature of the chirp generator to improve the RF-spectrum of the transmitted chirps. Causes transmission with 0° and 180° phase shift. 0: Disable (phase shift is always 0°) 1: Enable
3	WO brst BbClk	**CsqUse4Phases**: *Use Four Phases* Provides an additional 90° and 270° phase shift, along with the 0° and 180° phases enabled by CsqUsePhaseShift. 0: Disable 1: Enable
4	WO brst BbClk	**CsqAsyMode**: *Set Asymmetric Mode* When enabled, the Chirp Sequencer RAM is addressed asymmetrically. 0: Disable (access symmetrically: address increment AND decrement) 1: Enable (access increment only)
5	WO brst strb BbClk	**CsqMemAddrInit**: *Initialize Chirp Sequencer Memory Address* Sets the address counter of the chirp generator memory back to 0. Required before writing to a Chirp Sequencer RAM column. 0: Disable 1: Enable
6	WO brst BbClk	**CsqUseRam**: *Use Chirp Sequencer RAM* Enables the Chirp Sequencer RAM. 0: Disable (default matrix is used) 1: Enable

26.2.52 0x3F – Correlator controls

Used for setting the digital correlator in either FIX or MAP mode as well as for enabling POM detection mode.

Offset	RW	7	6	5	4	3	2	1	0
0x3F	R								
	init	0	0	0	0	0	0	0	0
	W	Internal Use Only				D3lPomLen		D3lPom En	D3lFixn Map
	init	0	0	0	0	0	0	0	1

Table 107. Field properties for register 0x3F

Bits	Prop	Description
0	WO brst BbClk	**D3lFixnMap**: *Use FIX or MAP Mode* Sets either FIX or MAP mode in the Correlator. FIX mode is default. 0: Use MAP mode 1: Use FIX mode (Default) Note: MAP mode (D3lFixnMap = 0) only works when FdmaEnable is disabled (set to 0), that is 80 MHz only. When using 22 MHz (FdmaEnable = 1), FIX mode (D3lFixnMap = 1) must be used.
1	WO brst BbClk	**D3lPomEn**: *Enable POM Mode* Enables the mean value calculation in the Correlator. 0: Disable 1: Enable Note: Can only be used in FIX mode.
2-3	WO brst BbClk	**D3lPomLen**: *Set POM Length Values* Select if the last 2, 4, 8, or 16 values should be used for calculating the mean value. 00: 2 01: 4 10: 8 11: 16

26.2.53 0x40 – Frame synchronization and bit synchronization

Switch back from MAP or FIX threshold1 to FIX threshold 0 detection in bitsync state when the in LeaveMapThresh1InBitsync programmed number of bits has passed without the transceiver has reached FrameSync state. The selection between MAP and FIX Threshold detection is programmed via Register D3lFixnMap.

UseMapThresh1InFramesync is used to select the detection mode after the FrameSync state is reached.

Offset	RW	7	6	5	4	3	2	1	0
0x40	R								
	init	0	0	0	0	0	0	0	0
	W	UseMap Thresh1 InFrame sync	LeaveMapThresh1InBitsync						
	init	0	0	0	0	0	0	1	1

Table 108. Field properties for register 0x40

Bits	Prop	Description
0-6	WO brst BbClk	**LeaveMapThresh1InBitsync**: *Switch Back to FIX Mode with Threshold 0* Switch back to FIX mode using correlator threshold 0 in bit synchronization state after programmed number of detected pulses. Default: 0x3

Table 108. Field properties for register 0x40 (continued)

Bits	Prop	Description
7	WO brst BbClk	**UseMapThresh1InFramesync**: *Use MAP Mode or FIX mode with Threshold 1 in Frame Sync State* Switch to MAP mode or FIX mode using correlator threshold 1 in frame synchronization state. 0: FIX with Threshold0 1: FIX with Threshold1 or MAP (D3LFixnMap)

26.2.54 0x41 – Bit synchronization

Used for setting the number of detected pulses (in bit synchronization state) before switching to either MAP mode or FIX mode using correlator threshold 1.

Offset	RW	7	6	5	4	3	2	1	0
0x41	R								
	init	0	0	0	0	0	0	0	0
	W	Internal Use Only	Go2MapThresh1InBitsync						
	init	0	0	0	0	0	1	1	1

Table 109. Field properties for register 0x41

Bits	Prop	Description
0	WO brst BbClk	**Go2MapThresh1InBitsync**: *Switch to FIX Mode with Threshold1 or MAP Mode* Switch from FIX mode using correlator threshold 0 to either MAP mode or FIX mode using correlator threshold 1 (D3LFixnMap) after programmed number of detected pulses in bit synchronization state. Default: 0x7

26.2.55 0x42 – Local oscillator, chirp sequencer and external PA controls

Used for enabling the Local Oscillator, the LO divider, the chirp sequencer clock, and the external power amplifier control pin.

Offset	RW	7	6	5	4	3	2	1	0
0x42	R								
	init	0	0	0	0	0	0	0	0
	W	Internal Use Only	Internal Use Only	Internal Use Only	Enable ExtPA	Invert RxClock	Enable CsqClock	Enable LOdiv10	Enable LO
	init	0	0	0	0	0	0	0	0

Table 110. Field properties for register 0x42

Bits	Prop	Description
0	WO brst BbClk	**EnableLO**: *Enable RF Local Oscillator* Enables the Local Oscillator. 0: Disable 1: Enable Note: Required only for tuning the Local Oscillator and for programming the Chirp Sequencer (CSQ) RAM.
1	WO brst BbClk	**EnableLOdiv10**: *Enable LO Frequency Divide By 10 Function* When enabled, the LO frequency is divided by 10 and delivered to digital part. 0: Disable 1: Enable Note: Required only for tuning the Local Oscillator and for programming the Chirp Sequencer (CSQ) RAM.
2	WO brst BbClk	**EnableCsqClock**: *Enable Chirp Sequencer Clock* Enables the Chirp Sequencer (CSQ) clock. 0: Disable 1: Enable Note: Required only for programming the Chirp Sequencer (CSQ) RAM.
3	WO brst BbClk	**InvertRxClock**: *Changing Sampling Edge for Receiver* It is recommended that this not be changes. Use the default value. 0: Disable 1: Enable
4	WO brst BbClk	**EnableExtPA**: *Enable External Power Amplifier Output* Enables the external power amplifier control pin to allow the use of an external power amplifier. It must be set when using an external power amplifier 0: Deactivate pin 1: Activate pin Note: The pin is low-active during transmit cycles and has high-impedance during non-transmit cycles.

26.2.56 0x43 – RF Rx low noise amplifier and TX power amplifier bias

Used for adjusting the frequency of the LNA response, and for adjusting the power amplifier bias current.

Offset	RW	7	6	5	4	3	2	1	0
0x43	R								
	init	0	0	0	0	0	0	0	0
	W		TxPaBias				LnaFreqAdjust		
	init	0	0	0	0	0	0	1	1

Table 111. Field properties for register 0x43

Bits	Prop	Description
0-2	WO brst BbClk	**LnaFreqAdjust**: *Adjust RF Receiver LNA Frequency* Frequency adjustment of the LNA (Low Noise Amplifier) frequency response (S21). 111b – Shift the middle frequency of the LNA frequency response to lower frequencies 000b – Shift the middle frequency of the LNA frequency response to higher frequencies Default: 0x3 (recommended)
4-6	WO brst BbClk	**TxPaBias**: *Adjust RF Transmitter Power Amplifier Bias* Adjusts the PA bias current to compensate process deviations. Default: 0x0 (recommended)

26.2.57 0x44 – Transmitter output power (Data, TimeB, and BrdCast)

Used for setting the RF transmitter output power for Data, TimeB (time beacon), and Brdcast (broadcast) packets.

Offset	RW	7	6	5	4	3	2	1	0
0x44	R								
	init	0	0	0	0	0	0	0	0
	W			TxOutputPower0					
	init	0	0	1	1	1	1	1	1

Table 112. Field properties for register 0x44

Bits	Prop	Description
0-5	WO brst BbClk	**TxOutputPower0**: *Adjust RF TX Output Power 0* Sets the transmitter output power for data (Data) packets, time beacon (TimeB) packets, and broadcast (BrdCast) packets. Default: 0x3F

26.2.58 0x45 – Transmitter output power (Ack, Req2S, and Clr2S)

Used for setting the RF Transmitter output power for Ack (acknowledgments), Req2S (request to send), and Clr2S (clear to send) packets.

Offset	RW	7	6	5	4	3	2	1	0
0x45	R								
	init	0	0	0	0	0	0	0	0
	W			TxOutputPower1					
	init	0	0	1	1	1	1	1	1

Table 113. Field properties for register 0x45

Bits	Prop	Description
0-5	WO brst BbClk	**TxOutputPower1**: *Adjust RF TX Output Power 1* Sets the transmitter output power for acknowledgement packets (Ack), request to send (Req2S) packets, and clear to send (Clr2S) packets. Default: 0x3F

26.2.59 0x46 – Quantization threshold for I

Used for programming the quantization threshold for the 5bit ADC for In-Phase value.

Offset	RW	7	6	5	4	3	2	1	0
0x46	R								
	init	0	0	0	0	0	0	0	0
	W				RfRxCompValueI				
	init	0	0	0	0	1	1	1	1

Table 114. Field properties for register 0x46

Bits	Prop	Description
0-4	WO brst BbClk	**RfRxCompValueI**: *Quantization Threshold for I* Default: 0xF

26.2.60 0x47 – Quantization threshold for Q

Used for programming the quantization threshold for the 5bit ADC for Quadrature-Phase value.

Offset	RW	7	6	5	4	3	2	1	0
0x47	R								
	init	0	0	0	0	0	0	0	0
	W				RfRxCompValueQ				
	init	0	0	0	0	1	1	1	1

Table 115. Field properties for register 0x47

Bits	Prop	Description
0-4	WO brst BbClk	**RfRxCompValueQ**: *Quantization Threshold for Q* Default: 0xF

26.2.61 0x48 – Symbol duration, symbol rate, and modulation system

Used for determining the transmission behavior, including setting the symbol duration and the symbol rate, as well as for setting a 2ary modulation system.

Offset	RW	7	6	5	4	3	2	1	0
0x48	R								
	init	0	0	0	0	0	0	0	0
	W	Internal Use Only	SymbolRate				SymbolDur		
	init	0	1	1	0	0	0	1	1

Table 116. Field properties for register 0x48

Bits	Prop	Description
0-2	WO BbClk	**SymbolDur**: *Set Symbol Duration* Set symbol duration between 500 ns and 4000 ns. – NA_SymbolDur500ns_C (0x0) – NA_SymbolDur1000ns_C (0x1) – NA_SymbolDur2000ns_C (0x2) – NA_SymbolDur4000ns_C (0x3) Default: 0x3
4-6	WO BbClk	**SymbolRate**: *Set Symbol Rate* Sets the symbol rate between 12 k Symbols and 2M Symbols – NA_SymbolRate500kSymbols_VC_C (0x0) – NA_SymbolRate1MSymbols_VC_C (0x1) – NA_SymbolRate2MSymbols_VC_C (0x2) – NA_SymbolRate125kSymbols_VC_C (0x6) – NA_SymbolRate250kSymbols_VC_C (0x7) Default: 0x7

26.2.62 0x49 – CRC2, CRC2 encryption, FEC, and encryption clock mode

Used for selecting the CRC2 type and for enabling the CRC2 encryption. It is also used for selecting the encryption/decryption clock mode and for enabling FEC.

Note: For more information on CRC types, see Section 19.2: Cyclic redundancy check (CRC) on page 86.

Offset	RW	7	6	5	4	3	2	1	0
0x49	R								
	init	0	0	0	0	0	0	0	0
	W	TxRxCryptClkMode				TxRx CryptCrc2 Mode	Use Fec	Crc2Type	
	init	0	0	0	0	0	0	0	0

Table 117. Field properties for register 0x49

Bits	Prop	Description
0-1	WO brst BbClk	**Crc2Type**: *Select Transceiver CRC2 Type* Selects the CRC2 used for reception and transmission. Cyclic Redundancy Checking and generation is calculated over the data field. Select CRC Type 1 (ISO/IEC3309): NA_Crc2Type1_VC_C (0x0) NA_Crc2Type1Bits_IC_C (0x10) Select CRC Type 2 (IEC 60870-5-1): NA_Crc2Type2_VC_C (0x1) NA_Crc2Type2Bits_IC_C (0x10) Select CRC Type 3 (CCITT-32): NA_Crc2Type3_VC_C (0x2) NA_Crc2Type3Bits_IC_C (0x20) 0 This register must be defined before: TxCmdStart = True or RxCmdStart = True Usage: Auto mode only.
2	WO brst BbClk	**UseFec**: *Enable Forward Error Correction (FEC)* Enables the forward error correction. Error correction is applied to the complete MAC frame using a (7,4) block code. – NA_UseFecOff_BC_C (0x0) Disables FEC. – NA_UseFecOn_BC_C (0x1) Enables FEC. This register must be defined before: TxCmdStart = True or RxCmdStart = True Usage: Auto mode only.
3	WO brst BbClk	**TxRxCryptCrc2Mode**: *Enable Encryption/Decryption CRC2 Mode* Enables or disables the Encryption/Decryption of CRC2. – NA_TxRxCryptCrc2ModeUncrypted_BC_C(0x0) In this mode, the CRC2 is calculated on the unencrypted data, then appended and both the CRC2 and Data field are encrypted. The advantage of this method is that integrity checking is possible. The disadvantage is that reception of the packet is not possible without the key. No key means no reception (no forwarding). – NA_TxRxCryptCrc2ModeEncrypted_BC_C(0x1) In this mode, the CRC2 is calculated on the encrypted data, then the CRC2 is appended unencrypted to the encrypted Data field. The advantages of this method is that no key for data error checking is needed and Forwarding is possible. The disadvantage is that no integrity checking is possible. This register must be defined before: TxCmdStart = True or RxCmdStart = True Usage: Auto mode only.

Table 117. Field properties for register 0x49 (continued)

Bits	Prop	Description
4-7	WO brst BbClk	**TxRxCryptClkMode**: *Set Encryption/Decryption Clock Mode* Selects the mode of the clock input value of the stream cipher for each CryptId. Bit4 belongs to CryptId0, .. Bit 7 belongs to CryptId3. – NA_TxRxCryptClkModeCryptClock_BC_C (0x0) Uses the RamTxCryptClock / RamRxCryptClock values for the clock input value calculating the en-/decryption key. – NA_TxRxCryptClkModeScrambInit_BC_C (0x1) The lowest 7 bits of the crypt clock (TxCryptClock /RxCryptClock are replaced by the value of the ScrambInit field. The MSBs are taken from RamTxCryptClock / RamRxCryptClock registers. For TX, the CryptID is taken from the TxCryptID field while for RX, the CryptID is taken from the received packet. This register must be defined before: TxCmdStart = True or RxCmdStart = True Usage: Auto mode only.

26.2.63 0x4A – Baseband buffer and MACFrame configuration

Used for setting the Baseband buffer mode (Duplex or Simplex) and the MACFrame mode (Auto or Transparent). It is also used for enabling FDMA which uses 22 MHz bandwidth and for swapping RX and TX baseband buffers.

Offset	RW	7	6	5	4	3	2	1	0
0x4A	R								SwapBb Buffers
	init	0	0	0	0	0	0	0	
	W	TxRx Mode			Fdma Enable		TxRxBb Buffer Mode0	TxRxBb Buffer Mode1	
	init	0	0	0	1	0	0	0	0

Table 118. Field properties for register 0x4A

Bits	Prop	Description
0	RW brst BbClk	**SwapBbBuffers**: *Exchange RX and TX Baseband Buffers* Exchanges the buffers between the baseband transmitter and receiver. When enabled, if a station is set up as an intermediate station (IS), the station does not have to move data from the RX buffer to the TX buffer. 0: Default. Data in buffers belong to their intended direction. 1: Enable data swapping (performs data buffer swapping). Data for the transmit operation are taken from the receive buffer, while data of the receive operation are stored in transmit buffer. Note: This is useful only in Duplex mode. See also: TxRxBbBufferMode1

Table 118. Field properties for register 0x4A (continued)

Bits	Prop	Description
1	WO brst BbClk	**TxRxBbBufferMode1**: *Baseband Buffer Mode 1: Duplex or Simplex* Sets the data buffer for TX and RX as either two blocks (Duplex) or a single block (Simplex). – NA_TxRxBufferMode1Duplex_BC_C (0x0) Sets Duplex mode In this mode, the data buffer is split into two blocks, one for TX and one for RX. TX and RX buffer operations are independent of each other. – NA_TxRxBufferMode1Simplex_BC_C (0x1) Sets Simplex mode. In this mode, one complete buffer is used either for TX or RX. Note: Buffer size and configuration depend on the combination of TxRxBbBufferMode0 and TxRxBbBufferMode1.
2	WO brst BbClk	**TxRxBbBufferMode0**: *Baseband Buffer Mode 0: Auto or Transparent* Sets the TX and RX data as either two segments (Auto mode where 256 bytes are allocated for buffers) or all four segments (Transparent mode where 512 bytes are allocated for buffers). – NA_TxRxBufferMode0Auto_BC_C (0x0) Sets Auto mode: TX/RX buffer size = 256 bytes. In this mode, the upper half of the Baseband RAM (two segments totalling 256 bytes) is used for data, as a TX and a RX buffer. The other two segments totalling 256 bytes are reserved for the following: • MAC frame header data • Station addresses • Encryption keys and related control values • RTC snapshots – NA_TxRxBufferMode0Transparent_BC_C (0x1) In this mode, the complete Baseband RAM (all four segments totalling 512 bytes) is used for data, as either a TX or a RX buffer (Simplex mode) or one TX and one RX buffer (Duplex mode). In Duplex mode, TX and RX operations are independant of each other. Note: Buffer size and configuration depend on the combination of TxRxBbBufferMode0 and TxRxBbBufferMode1.
4	WO brst BbClk	**FdmaEnable**: *Enables FDMA* Enables FDMA (Frequency Division Multiple Access), which uses 22 MHz bandwidth. When disabled, 80 MHz bandwidth is used. 0: Disable 1: Enable (Default)

26.2.64 0x4B to 0x4C – Transceiver chirp matrix

In a 2ary system, each bit (whose possible values are either 0 or 1) is represented as a chirp pulse. ChirpMatrix0 selects the type of chirp pulse which represents the value 0. ChirpMatrix1 selects the type of chirp pulse that represents the value 1.

The pulse types include Upchirp, Downchirp, Folded Minus Pulse, Plus Folded Pulse, and Off Chirp. See also *Table 119: Pulse types on page 174*.

Offset	RW	7	6	5	4	3	2	1	0
0x4B	R								
	init	0	0	0	0	0	0	0	0
	W		ChirpMatrix1				ChirpMatrix0		
	init	0	0	0	1	0	0	0	0
0x4C	R								
	init	0	0	0	0	0	0	0	0
	W		ChirpMatrix3				ChirpMatrix2		
	init	0	1	0	0	0	0	1	1

Pulse Types

For each of ChirpMatrix0, ChirpMatrix1, ChirpMatrix2, and ChirpMatrix3, select one of the pulse types as listed in the following table.

Table 119. Pulse types

To Set Pulse Type	Use This Constant	Hex Value
Downchirp	NA_ChirpDown_VC_C	0x0
Upchirp	NA_ChirpUp_VC_C	0x1
Minus Folded Chirp	NA_ChirpFoldMinus_VC_C	0x2
Plus Folded Chirp	NA_ChirpFoldPlus_VC_C	0x3
Off Chirp	NA_ChirpOff_VC_C	0x4

Table 120. Field properties for registers 0x4B and 0x4C

Bits	Prop	Description
0-2	WO brst BbClk	**ChirpMatrix0**: *Set Pulse Type for Values 0* Select pulse type (see *Table 119*) This register must be defined before: TxCmdStart = True or RxCmdStart = True
4-6	WO brst BbClk	**ChirpMatrix1**: *Set Pulse Type for Value 1* Select pulse type (see *Table 119*) This register must be defined before: TxCmdStart = True or RxCmdStart = True

26.2.65 0x4D – TX underrun, CIFS waiting, and preamble/tail sequences

Used for setting the value of symbol 0 or 1 in the MACFrame Preamble and Tail. It is also used for enabling the aborting of a transmission through the use of the TxCmdStop command, as well as for Disabling waiting for CIFS (Common InterFrame Switching time). I

Note: Chirp pulses are transmitted in the following sequence:

- *TxPreTrailMatrix0*
- *TxPreTrailMatrix1*
- *TxPreTrailMatrix0*
- *TxPreTrailMatrix1*
- *and so on.*

An Off symbol cannot be used in the Preamble or the Tail. Therefore, in an Up/Off or Down/Off modulation system, both values must be the same value.

Offset	RW	7	6	5	4	3	2	1	0
0x4D	R								
	init	0	0	0	0	0	0	0	0
	W	TxMac CifsDis			TxUnder runIgnore	TxPreTrailMatrix1		TxPreTrailMatrix0	
	init	0	0	0	1	0	1	0	0

Table 121. Field properties for register 0x4D

Bits	Prop	Description
0-1	WO brst BbClk	**TxPreTrailMatrix0**: *Value for Symbol 0 in Preamble and Tail (Trail)* Sets the value of symbol 0 in the MACFrame Preamble and Tail. – Symbol used in TxRxChirpMatrix0 (0x0) – Symbol used in TxRxChirpMatrix1 (0x1) – Symbol used in TxRxChirpMatrix2 (0x2) – Symbol used in TxRxChirpMatrix3 (0x3) This register must be defined before: TxCmdStart = True or RxCmdStart = True Usage: Auto and transparent mode.
2-3	WO brst BbClk	**TxPreTrailMatrix1**: *Value for Symbol 1 in Preamble and Tail (Trail)* Sets the value of symbol 1 in the MACFrame Preamble and Tail. – Symbol used in TxRxChirpMatrix0 (0x0) – Symbol used in TxRxChirpMatrix1 (0x1) – Symbol used in TxRxChirpMatrix2 (0x2) – Symbol used in TxRxChirpMatrix3 (0x3) This register must be defined before: TxCmdStart = True or RxCmdStart = True Usage: Auto and transparent mode.
4	WO brst BbClk	**TxUnderrunIgnore**: *TX Underrun Ignore/Auto-Abort* When a transmission underrun occurs, if this bit is enabled, the underrun error will be ignored and the transmission can continue or software can abort the transmission. If this bit is disabled, transmission is aborted automatically when a transmission underrun occurs. 0: Disable. Transmission aborts automatically. 1: Enable. Transmission does not abort automatically. It must be aborted by the command TxCmdStop. (Default)
7	WO brst BbClk	**TxMacCifsDis**: *Disable Waiting for CIFS (Common InterFrame Switching time)* When waiting for an idle media for transmission the transmitter first waits for CIFS time (time needed of the station for switching from transmit to receive mode). When CSMA is enabled, several back-off slots can occur after CIFS. Using this register, waiting for CIFS time can be disabled. 0: Enable waiting 1: Disable waiting

26.2.66 0x4E – Media access control for transmitter

Used for enabling virtual carrier sensing (with or without Ack packets), as well for enabling physical carrier sensing using either RSSI or chirp detection. It is also used for enabling ARQ, which uses acknowledgements, a three-way handshake mechanism (to avoid collisions and eliminate “hidden node” problems), and a back-off mechanism for fair medium access.

Offset	RW	7	6	5	4	3	2	1	0
0x4E	R								
	init	0	0	0	0	0	0	0	0
	W	TxFrag Prio	TxBack OffAlg	Tx3Way	TxArq	TxVCarr SensAck	TxPhCarr SenseMode		TxVCarr Sens
	init	0	0	0	1	0	0	0	0

Table 122. Field properties for register 0x4E

Bits	Prop	Description
0	WO brst BbClk	**TxVCarrSens**: *Virtual Carrier Sensing* Virtual Carrier Sensing calculates the end of the current transmission based on information taken from the received header. When this carrier sense method is enabled, the NAV (Network Allocation Vector) is calculated using the packet type and Length field of received packets. Also, the transmitter suspends medium access until the NAV expires. – NA_TxVCarrSensOff_BC_C (0x0) – NA_TxVCarrSensOn_BC_C (0x1) This register must be defined before: TxCmdStart = True Usage: Auto mode only.
1-2	WO brst BbClk	**TxPhCarrSenseMode**: *Physical Carrier Sensing Mode: RSSI or Chirp Detection* Physical carrier sensing uses either RSSI or chirp symbol detection. Using RSSI, if AgcGain is lower than AgcRssiThreshold, then the carrier is busy. Using the other method, if a chirp is detected, then the carrier is busy. Physical carrier sensing mode can be based either on chirp detection, RSSI threshold, or both chirp detection and RSSI threshold. – NA_TxPhCarrSensModeOff_VC_C (0x0) Physical carrier sensing is not applied. – NA_TxPhCarrSensModeSymbols_VC_C (0x1) Physical carrier sensing is based on chirp symbol detection. – NA_TxPhCarrSensModeRssi_VC_C (0x2) Physical carrier sensing is based on an RSSI threshold. – NA_TxPhCarrSensModeSymbolsRssi_VC_C (0x3) Physical carrier sensing is based on both symbol detection and RSSI threshold. This register must be defined before: TxCmdStart = True. Usage: Auto and transparent mode.
3	WO brst BbClk	**TxVCarrSensAck**: *Include Ack Packets for Virtual Carrier Sense* When virtual carrier Sense is enabled and this bit is set the time of an Ack packet is taken into calculation when calculating the NAV (Network Allocation Vector). – NA_TxVCarrSensAckOff_BC_C (0x0) NAV calculation does not regard Ack packets. – NA_TxVCarrSensAckOn_BC_C (0x1) NAV calculation regards Ack packets. This register must be defined before: TxCmdStart = True. Usage: Auto mode only.

Table 122. Field properties for register 0x4E (continued)

Bits	Prop	Description
4	WO brst BbClk	**TxArq**: *TX ARQ (Automatic Repeat Request) Error Correction Scheme for Data Packets* When this bit is set, the station waits for an Ack packet as a response to a transmitted Data packet. – NA_TxArqOff_BC_C (0x0) ARQ is disabled. When ARQ is disabled, the transmitter does not wait for an Ack packet as a response to a Data packet. – NA_TxArqOn_BC_C (0x1) ARQ is enabled. With ARQ, the transmitting station waits for an acknowledgement (ACK packet). The packet is retransmitted automatically if TxArqMax > 0. Note: ARQ must be disabled for Brdcast and TimeB packets. This register must be defined before: TxCmdStart = True. Usage: Auto mode only.
5	WO brst BbClk	**Tx3Way**: *Three-Way Handshake* The three-way handshake uses a Req2S and Clr2S mechanism. It is only used with Data packets. It is not used with Brdcast and TimeB packets. – NA_Tx3WayOff_BC_C (0x0) Three-way handshake is disabled. – NA_Tx3WayOn_BC_C (0x1) Three-way handshake is enabled. This register must be defined before: TxCmdStart = True. Usage: Auto mode only.
6	WO brst BbClk	**TxBackOffAlg**: *Back-Off Algorithm* The exponential back-off algorithm waits a specific period of time before reattempting a transmission. – NA_TxBackOffAlgOff_BC_C (0x0) Back-off algorithm is disabled. The complete seed is taken (see TxBackoffSeed) and the seed is not modified. – NA_TxBackOffAlgOn_BC_C (0x1) Back-off algorithm is enabled. The number of bits taken from seeds are increased by one when a transmission is not acknowledged.A random value of 3 bit size is taken for the first transmission. This register must be defined before: TxCmdStart = True. Usage: Auto mode only.
7	WO brst BbClk	**TxFragPrio**: *Fragmentation Prioritization* Enables fragmentation prioritization scheme used in the back-off time algorithm. 0: Disable. The MSB of the back-off counter is set with the MSB from the random value. 1: Enable. Sets the MSB of the Back-Off counter to value zero (higher priority). This register must be defined before: TxCmdStart = True. Usage: Auto mode only.

26.2.67 0x4F – Back-off seed for back-off time

Used for setting the back-off seed, which is either the start value of the random number generator for exponential back-off or the number of back-off slots which has to pass by before next media access will be attempted.

Offset	RW	7	6	5	4	3	2	1	0
0x4F	R								
	init	0	0	0	0	0	0	0	0
	W	TxBackOffSeed							
	init	0	0	0	0	0	0	0	0

Table 123. Field properties for register 0x4F

Bits	Prop	Description
0-7	WO brst BbClk	**TxBackOffSeed**: *Back-Off Seed* Sets either the start value of the random number generator for exponential back-off or the number of the back-off slots. – If TxBackOffAlg = TxBackOffAlgOn, then this sets the seed value for the pseudo-number generator. It can be set for every packet and improves the random properties of exponential Back-Off. – If TxBackOffAlg = TxBackOffAlgOff, then this sets the number of Back-Off slots before the next media access will start. It is not mandatory to set this register. Usage: Auto mode only.

26.2.68 0x50 – Transmission encryption

Used for resetting the internally stored values of the CryptSeqN bits (one for each key), which are then compared to the programmed TxCryptSeqN bit after a key has been selected and encryption starts. When the compared bits are unequal, the crypt clock used for encryption is incremented. This register is also used for enabling transmission encryption security and to select a secret key, as well as and for enabling a one bit sequential numbering scheme for encryption.

Offset	RW	7	6	5	4	3	2	1	0
0x50	R								
	init	0	0	0	0	0	0	0	0
	W	TxCrypt SeqN	TxCryptId		TxCrypt En	TxCryptSeqReset			
	init	0	0	0	0	0	0	0	0

Table 124. Field properties for register 0x50

Bits	Prop	Description
0-3	WO brst strb BbClk	**TxCryptSeqReset**: *TX Decryption Sequence Reset* Resets the last transmitted SeqN bit independently for each key. TxCryptSeqReset[TxCryptId] When one of these bits is set to 1, the internally stored CryptSeqN bit for the corresponding key is reset to 0. Note: The internally stored CryptSeqN bits (4 for TX and 4 for RX) are always zero after reset (/POnReset or Baseband Reset).

Table 124. Field properties for register 0x50 (continued)

Bits	Prop	Description
4	WO brst BbClk	**TxCryptEn**: *Encryption Switch* Enables the encryption security feature of the chip, which uses a 32 bit clock value, a 48 bit address, and a 128 bit secret key to generate the encryption cipher. This sets the CryptEn bit in the MACFrame. 0: Disable 1: Enable This register must be defined before: TxCmdStart = True Usage: Auto mode only.
5-6	WO brst BbClk	**TxCryptId**: *Secret Key Selection for Encryption* Selects one of four secret keys as well as one of the four TX crypt clocks, which are found in page 1 of the baseband RAM. It is only used for Data packets, Brdcast packets and TimeB packets as only data is encrypted and other packet types have no data fields. – Select first secret key (0x0) – Select second secret key (0x1) – Select third secret key (0x2) – Select fourth secret key (0x3) This register must be defined before: TxCmdStart = True Usage: Auto mode only.
7	WO brst BbClk	**TxCryptSeqN**: *Encryption Sequence Number* This bit is compared to the internally stored CryptSeqN bit (the TxCryptseqN bit of last transmission) of the corresponding key. If they are unequal then the used crypt clock is incremented before generating the cipher. This register must be defined before: TxCmdStart = True Usage: Auto mode only.

26.2.69 0x51 – Transmit scrambler

Used for enabling scrambling of TX packets and for initializing the bit scrambler.

Offset	RW	7	6	5	4	3	2	1	0
0x51	R								
	init	0	0	0	0	0	0	0	0
	W	Tx ScrambEn	TxScrambInit						
	init	1	1	1	1	1	1	1	1

Table 125. Field properties for register 0x51

Bits	Prop	Description
0-6	WO brst BbClk	**TxScrambInit**: *TX Scrambling Initialization* Initializes the value for the scrambling unit. Default: 0x7F Note: Do NOT program TxScrambInit with 0x00. There is no need to initialize the scrambler for each packet. This register must be defined before: TxCmdStart = True Usage: Auto mode only.

Table 125. Field properties for register 0x51 (continued)

Bits	Prop	Description
7	WO brst BbClk	**TxScrambEn**: *Transmission Scrambling* Enables scrambling in a transmitted MACFrame. 0: Disable. Value 0 is intended for debugging purposes only. 1: Enable. (Default) To randomize the data from highly redundant patterns, scrambling must be enabled during normal operations. This register must be defined before: TxCmdStart = True Usage: Auto mode only.

26.2.70 0x52 to 0x53 – TX data size in transparent mode

Used for setting the number of data bytes for transmission in Transparent mode.

Offset	RW	7	6	5	4	3	2	1	0
0x52	R								
	init	0	0	0	0	0	0	0	0
	W	TxTransBytes							
	init	0	0	0	0	0	0	0	0
0x53	R								
	init	0	0	0	0	0	0	0	0
	W				TxTransBytes				
	init	0	0	0	0	0	0	0	0

Table 126. Field properties for registers 0x52 and 0x53

Bits	Prop	Description
0-12	WO brst BbClk	**TxTransBytes**: *Number of Data Bytes in TX MACFrame in Transparent Mode* When the transceiver is set in Transparent mode, this sets the number of bytes of data in the completed MACFrame that is ready for transmission. These are all bytes between the Syncword and the Tail. This register must be defined before: TxCmdStart = True Usage: Transparent mode only.

26.2.71 0x54 – Packet type for auto mode and TX source address

Used for storing the type of packet that will be transmitted, and for selecting as the source address of the packet, which is either Station Address 0 (StaAddr0) or Station Address 1 (StaAddr1).

Offset	RW	7	6	5	4	3	2	1	0
0x54	R								
	init	0	0	0	0	0	0	0	0
	W	TxAddr Slct				TxPacketType			
	init	0	0	0	0	0	0	0	0

Table 127. Field properties for register 0x54

Bits	Prop	Description
0-3	WO brst BbClk	**TxPacketType**: *Packet Type For Transmission in Auto Mode* Stores the type of packet to be transmitted. Only Data packets, TimeB packets, and Broadcast packets can be selected. – NA_TypeCodeData_VC_C (0x0) – NA_TypeCodeTimeB_VC_C (0x2) – NA_TypeCodeBrdcast_VC_C (0x3) This register must be defined before: TxCmdStart = True Usage: Auto mode only.
7	WO brst BbClk	**TxAddrSlct**: *Source Address for Transmission* Selects one of the two programmed station addresses as source address for packets to be transmitted. 0: Station Address 1 (RamStaAddr0) 1: Station Address 2 (RamStaAddr1). This register must be defined before: TxCmdStart = True Usage: Auto mode only.

26.2.72 0x55 – Start/stop transmissions and TX buffer fill command

Used for starting and stopping the transmitter and for setting a command that indicates that the lower subblock and/or upper subblock of the TX buffer is ready for transmission.

Offset	RW	7	6	5	4	3	2	1	0
0x55	R								
	init	0	0	0	0	0	0	0	0
	W					TxBufferCmd		TxCmdStart	TxCmdStop
	init	0	0	0	0	0	0	0	0

Table 128. Field properties for register 0x55

Bits	Prop	Description
0	WO brst strb BbClk	**TxCmdStop**: *Stop Transmission* This interrupts the current transmission by aborting the transmission of a packet (in Auto mode) or stops/disables the transmission (in Transparent mode). It also resets the transmit buffer state. 0: Default 1: Stop transmission
1	WO brst strb BbClk	**TxCmdStart**: *Start Transmission* Starts the configured transmission. It initiates the transmission of a packet (in Auto mode) or starts the transmission of data (in Transparent mode). 0: Default 1: Start transmission

Table 128. Field properties for register 0x55 (continued)

Bits	Prop	Description
2-3	WO brst strb BbClk	**TxBufferCmd**: *Transmit Buffer Ready for Transmission* Indicates that either the lower subblock and/or the upper subblock of the TX Buffer is ready (filled) for transmission. – Lower subblock filled for transmission LSB – Upper subblock filled for transmission MSB This command must be written after the TX buffer has been written to with the data to be transmitted.

26.2.73 0x56 – Start/stop receptions and RX buffer fill command

Used for starting and stopping the receiver and for setting a command that indicates that the lower subblock and/or upper subblock of the RX buffer is empty and ready for a received packet.

Offset	RW	7	6	5	4	3	2	1	0
0x56	R								
	init	0	0	0	0	0	0	0	0
	W					RxBufferCmd		RxCmdStart	RxCmdStop
	init	0	0	0	0	0	0	0	0

Table 129. Field properties for register 0x56

Bits	Prop	Description
0	WO brst strb BbClk	**RxCmdStop**: *Stop Current Reception* This interrupts the current reception by aborting the reception of a packet (in Auto mode) or stops/disables the reception (in Transparent mode). It also resets the receive buffer state. 0: Default 1: Stop reception
1	WO brst strb BbClk	**RxCmdStart**: *Start Receiver in Configured Mode* Starts the reception of a packet (correlation in Auto or Transparent Mode). If a receive or transmit process is already active, the command is stored until the transceiver gets in the idle state. The command is then executed and reset. 0: Default 1: Start reception
2	WO brst strb BbClk	**RxBufferCmd**: *Read Receive Buffer* Indicates either the lower subblock and/or the upper subblock of the RX Buffer are ready (empty) for reception. – Lower subblock of RX buffer is read LSB – Upper subblock of RX buffer is read MSB

26.2.74 0x57 – Receiver decryption sequence reset

Used for resetting the internally stored values of the CryptSeqN bits (one for each key; received in the last packet using the same key) which are compared to the currently received RxCryptSeqN (part of the MAC header) bit. When the compared bits are unequal, then the crypt clock used for decryption is incremented before generating the cipher.

Offset	RW	7	6	5	4	3	2	1	0
0x57	R								
	init	0	0	0	0	0	0	0	0
	W					RxCryptSeqReset			
	init	0	0	0	0	0	0	0	0

Table 130. Field properties for register 0x57

Bits	Prop	Description
0-3	WO brst strb BbClk	**RxCryptSeqReset**: *RX Decryption Sequence Reset* Resets the last received SeqN bit independently for each key. RxCryptSeqReset[RxCryptId] When one of these bits is set to 1, the internally stored CryptSeqN bit for the corresponding key is reset to 0. Note: The internally stored CryptSeqN bits (4 for TX and 4 for RX) are always zero after reset (/POnReset or Baseband Reset).

26.2.75 0x58 to 0x59 – RX data size in transparent mode

Used for setting the number of bytes to receive in Transparent mode.

Offset	RW	7	6	5	4	3	2	1	0
0x58	R								
	init	0	0	0	0	0	0	0	0
	W	RxTransBytes							
	init	0	0	0	0	0	0	0	0
0x59	R								
	init	0	0	0	0	0	0	0	0
	W				RxTransBytes				
	init	0	0	0	0	0	0	0	0

Table 131. Field properties for registers 0x58 to 0x59

Bits	Prop	Description
0-12	WO brst BbClk	**RxTransBytes**: *Number of Data Bytes in Received MACFrame in Transparent Mode* Because the complete MACFrame is interpreted as a Data field Transparent mode and no information exists about how many bytes will follow, the number of bytes the chip should receive must be programmed before these packets can be received. These are all bytes between the Syncword and the Tail. Note: A value of all zeros is interpreted as 8192 bytes.

26.2.76 0x5A – RX ES and IS address matching, CRC1, CRC2, and ARQ

Used for enabling address matching for End Stations and Intermediate Stations, for enabling CRC1 checking on TimeB packets, and for enabling CRC2 checking on received Data, TimeB and Broadcast packets. It is also used for configuring the format of a segmented address by setting the length of the DeviceID.

Note: For more information on CRC types, see Section 19.2: Cyclic redundancy check (CRC) on page 86.

Offset	RW	7	6	5	4	3	2	1	0
0x5A	R								
	init	0	0	0	0	0	0	0	0
	W	RxAddrSegDevIdL		RxAddr SegIsMode	RxAddr SegEsMode	RxArqMode		RxCrc2 Mode	RxTimeB Crc1Mode
	init	0	0	0	0	1	0	1	1

Table 132. Field properties for register 0x5A

Bits	Prop	Description
0	WO brst BbClk	**RxTimeBCrc1Mode**: *CRC1 Mode for TimeB Packets* TimeB packets that have failed CRC1 checks can either be received by ignoring the failed CRC1 check, or a failed CRC1 check will cause the reception to terminate. The status of the CRC1 check is reported in the register RxCrc1Stat. – NA_RxTimeBCrc1ModeOff_BC_C (0x0) Failed CRC1 check terminates reception. – NA_RxTimeBCrc1ModeOn_BC_C (0x1) Receive TimeB packets and ignore failed CRC1 check. (Default)
1	WO brst BbClk	**RxCrc2Mode**: *CRC2 Mode for Data, TimeB, and Broadcast Packets* Data, TimeB, and Broadcast packets that have failed CRC2 checks can be received by ignoring the failed CRC2 check. – NA_RxCrc2ModeTrigOff_BC_C (0x0) Ignore failed CRC2 check and trigger RxEnd interrupt. Receiver is stopped. – NA_RxCrc2ModeTrigOn_BC_C (0x1) RxEnd is only triggered when CRC2 check succeeds. When the check fails the receiver remains active to receive the next packet. When overflow occurs this will be ignored. (Default)
2-3	WO brst BbClk	**RxArqMode**: *RX ARQ (Automatic Repeat Request) Mode for Data Packets* Selects the ARQ mode, including: – Ack should be sent as a reply to a received Data packet – Ack should not be sent when a Data packet is received – Ack should be sent when a CRC1 check succeeds – Ack should be sent when both a CRC1 and a CRC2 check succeeds. – NA_RxArqModeNone_VC_C (0x0) ARQ is disabled and an Ack packet is not sent in response to a Data packet. – NA_RxArqModeCrc1_VC_C (0x1) The ARQ scheme is based on the CRC1 only. An Ack packet is sent in response to a Data packet only if CRC1 and address matching succeeded. – NA_RxArqModeCrc2_VC_C (0x2) The ARQ scheme is based on the CRC1 and CRC2. An Ack packet is sent in response to a Data packet only if CRC1, CRC2 , and address matching succeeded. (Default) This register must be defined before: RxCmdStart = True Usage: Auto mode only.

Table 132. Field properties for register 0x5A (continued)

Bits	Prop	Description
4	WO brst BbClk	**RxAddrSegEsMode**: *End Station (ES) Address Matching* Enables the address matching scheme for End Stations. Address matching for the destination station address as set in RamStaAddr1 is automatically disabled. ES matching succeeds when ES is enabled and the following is true: – IG/UL bits = RamStaAddr1 – ISubnetId = 0 – SubnetID = RamStaAddr1 – DeviceID = RamStaAddr1 0: Disable 1: Enable
5	WO brst BbClk	**RxAddrSegIsMode**: *Intermediate Station (IS) Address Matching* Enables the address matching scheme for Intermediate Stations. IS matching succeeds when IS is enabled and the following is true: – IG/UL bits = RamStaAddr1 – ISubnetId = RamStaAddr1 0: Disable 1: Enable
6-7	WO brst BbClk	**RxAddrSegDevIdL**: *Segmented Address Format* Configures the format of a segmented address by defining the length of the Device ID. Possible lengths include 2, 6, 10, and 14 bits. A segmented address is always 48 bits and can be configured as follows: **Device ID** / **Subnet ID** / **ISubnet ID** / **IG/UL** 2 / 22 / 22 / 2 6 / 20 / 20 / 2 10 / 18 / 18 / 2 14 / 16 / 16 / 2 – Device ID = 2 bits (0X0) – Device ID = 6 bits (0X1) – Device ID = 10 bits (0X2) – Device ID = 14 bits (0X3)

26.2.77 0x5B – RX Data, Brdcast, TimeB, address mode, and ranging pulses

Used for enabling the reception of Data packets, Broadcast packets, and TimeB packets. It is also used for setting the address mode for Data packets, as either Address Matching Mode or Promiscuous Mode. It is also used for setting the number of up and down pulses, which are used for determining the ToA average when performing ranging measurements.

Offset	RW	7	6	5	4	3	2	1	0
0x5B	R								
	init	0	0	0	0	0	0	0	0
	W	RangingPulses				RxAddr Mode	RxTimeB En	RxBrdcast En	RxDataEn
	init	0	1	0	1	1	1	1	1

Table 133. Field properties for register 0x5B

Bits	Prop	Description
0	WO brst BbClk	**RxDataEn**: *Data Packet Reception* Enables the reception of Data packets. – NA_RxDataOff_BC_C (0x0) Data packets are discarded. – NA_RxDataOn_BC_C (0x1) Data packets are received. (Default) This register must be defined before: RxCmdStart = True Usage: Auto mode only.
1	WO brst BbClk	**RxBrdcastEn**: *Broadcast Packet Reception* Enable the reception of Broadcast packets. – NA_RxBrdcastOff_BC_C (0x0) Broadcast packets are discarded. – NA_RxBrdcastOn_BC_C (0x1) Broadcast packets are received. This register must be defined before: RxCmdStart = True Usage: Auto mode only.
2	WO brst BbClk	**RxTimeBEn**: *TimeB Packet Reception* Enables reception of TimeB packets. – NA_RxTimeBOff_BC_C (0x0) TimeB packets are discarded. – NA_RxTimeBOn_BC_C (0x1) TimeB packets are received. This register must be defined before: RxCmdStart = True Usage: Auto mode only.
3	WO brst BbClk	**RxAddrMode**: *Address Mode for Data Packets: Address Matching Mode or Promiscuous Mode* Sets the mode of address comparison. Data packets can be received independent of address matching (Promiscuous Mode), or with address matching (Address Matching Mode). – NA_RxAddrModeOff_BC_C (0x0) Sets Promiscuous Mode. The complete Data packet is received and receive interrupts are triggered independent of the address set in the received Data packet. – NA_RxAddrModeOn_BC_C (0x1) Sets Address Matching Mode. A Data packet is received and receive interrupts are triggered only if the destination address set in the received Data packet matches the receiver station address. (Default) This register must be defined before: RxCmdStart = True Usage: Auto mode only.

Table 133. Field properties for register 0x5B (continued)

Bits	Prop	Description
4-7	WO brst BbClk	**RangingPulses**: *Number of Up and Down Pulses Used for ToA Measurement Average* Sets the number of up and down ranging pulses used for determining the ToA (Time of Arrival) average measurement. – NA_RangingPulses1_VC_C (0x0) Use 1 up and 1 down pulse for average. – NA_RangingPulses2_VC_C (0x1) Use 2 up and 2 down pulses for average. – NA_RangingPulses4_VC_C (0x2) Use 4 up and 4 down pulses for average. – NA_RangingPulses8_VC_C (0x3) Use 8 up and 8 down pulses for average. – NA_RangingPulses16_VC_C (0x4) Use 16 up and 16 down pulses for average. – NA_RangingPulses24_VC_C (0x5) Use 24 up and 24 down pulses for average. (Default) – NA_RangingPulses32_VC_C (0x6) Use 32 up and 32 down pulses for average. – NA_RangingPulses40_VC_C (0x7) Use 40 up and 40 down pulses for average. – NA_RangingPulses48_VC_C (0x8) Use 48 up and 48 down pulses for average. – NA_RangingPulses56_VC_C (0x9) Use 56 up and 56 down pulses for average. – NA_RangingPulses64_VC_C (0xA) Use 64 up and 64 down pulses for average.

26.2.78 0x5C – Delay detection tuning

Used for tuning the delay in the correlator.

Offset	RW	7	6	5	4	3	2	1	0
0x5C	R								
	init	0	0	0	0	0	0	0	0
	W				PulseDetDelay				
	init	0	0	0	0	0	1	0	1

Table 134. Field properties for register 0x5C

Bits	Prop	Description
0-4	WO brst BbClk	**PulseDetDelay**: *Delay Tuning For Pulse Detection* Tunes the delay in the correlator. Default: 0x5

26.2.79 0x5D – Receive bit detector controls (bit detectors)

Used for disabling the detection of up pulses (Upchirps) and/or down pulses (Downchirps), and for setting the required number of consecutive pulses needed to be outside of the gate center before a readjustment of the gate occurs.

Offset	RW	7	6	5	4	3	2	1	0
0x5D	R								
	init	0	0	0	0	0	0	0	0
	W			UpPulse DetectDis	DownPulse DetectDis		GateAdjThreshold		
	init	0	0	0	0	0	1	1	1

Table 135. Field properties for register 0x5D

Bits	Prop	Description
0-2	WO brst BbClk	**GateAdjThreshold**: *Gate Adjust Threshold* Because of clock drifting, the detected pulse can run out of center of the gate. This sets the required number of consecutive pulses outside of the gate center before a readjustment of the gate takes place. Default: 0x7 This register must be defined before: RxCmdStart = True
4	WO brst BbClk	**DownPulseDetectDis**: *Disable Detection of Down Pulses* Disables the detection of down pulses (Downchirps) in the bit detector. 0: Enable 1: Disable
5	WO brst BbClk	**UpPulseDetectDis**: *Disable Detection of Up Pulses* Disables the detection of up pulses (Upchirps) in the bit detector. 0: Enable 1: Disable

26.2.80 0x5E – Bit detection gate size and adjustment

Used for setting the size of the bit detection gate for unsynchronized state, for bit synchronization state, and for frame synchronized state. It is also used for enabling the adjustment of the detection gate in bit synchronization state and frame synchronization state.

Offset	RW	7	6	5	4	3	2	1	0
0x5E	R								
	init	0	0	0	0	0	0	0	0
	W	GateAdj Frame syncEn	GateAdj Bitsync En	GateSizeFramesync		GateSizeBitsync		GateSizeUnsync	
	init	1	1	0	1	0	1	0	1

Table 136. Field properties for register 0x5E

Bits	Prop	Description
0-1	WO brst BbClk	**GateSizeUnsync**: *Gate Size for Unsynchronized State* Sets the size of the detection gate in the baseband clock cycles for the unsynchronized state. – NA_GateSize3Slots_VC_C (0x0) Sets gate size = 3 slots. – NA_GateSize5Slots_VC_C (0x1) Sets gate size = 5 slots. (Default) – NA_GateSize7Slots_VC_C (0x2) Sets gate size = 7 slots. – NA_GateSize9Slots_VC_C (0x3) Sets gate size = 9 slots. Note: 1 slot = 31.25 ns (one baseband clock cycle). This register must be defined before: RxCmdStart = True
2-3	WO brst BbClk	**GateSizeBitsync**: *Gate Size for Bit Synchronization State* Sets the size of the detection gate in baseband clock cycles for the bit synchronization state. – NA_GateSize3Slots_VC_C (0x0) Sets gate size = 3 slots. – NA_GateSize5Slots_VC_C (0x1) Sets gate size = 5 slots. (Default) – NA_GateSize7Slots_VC_C (0x2) Sets gate size = 7 slots. – NA_GateSize9Slots_VC_C (0x3) Sets gate size = 9 slots. Note: 1 slot = 31.25 ns (one baseband clock cycle). This register must be defined before: RxCmdStart = True
4-5	WO brst BbClk	**GateSizeFramesync**: *Gate Size for Frame Synchronization State* Sets the size of the detection gate in baseband clock cycles for the frame synchronization state. – NA_GateSize3Slots_VC_C (0x0) Sets gate size = 3 slots. – NA_GateSize5Slots_VC_C (0x1) Sets gate size = 5 slots. (Default) – NA_GateSize7Slots_VC_C (0x2) Sets gate size = 7 slots. – NA_GateSize9Slots_VC_C (0x3) Sets gate size = 9 slots. Note: 1 slot = 31.25 ns (one baseband clock cycle). This register must be defined before: RxCmdStart = True
6	WO brst BbClk	**GateAdjBitsyncEn**: *Detection Gate Adjustment for Bit Synchronization State* Enables gate adjustment in bit synchronization state. 0: Disable 1: Enable (Default)

Table 136. Field properties for register 0x5E (continued)

Bits	Prop	Description
7	WO brst BbClk	**GateAdjFramesyncEn**: *Detection Gate Adjustment for Frame Synchronization State* Enables gate adjustment in frame synchronization state. 0: Disable 1: Enable (Default) Note: GateAdjFramesyncEn must be disabled for Ranging.

26.2.81 0x5F – Bit synchronization/unsynchronization thresholds

Used for setting the threshold (number of consecutive detected chirp pulses) to enter bit synchronization state from the unsynchronized state, as well as for setting the threshold (number of consecutive lost chirp pulses) to enter the unsynchronized state from bit synchronization state.

Note: The frame synchronization state is reached after the SyncWord has been received.

Offset	RW	7	6	5	4	3	2	1	0
0x5F	R								
	init	0	0	0	0	0	0	0	0
	W		LeaveBitsyncThreshold				Go2BitsyncThreshold		
	init	0	1	1	0	0	0	1	0

Table 137. Field properties for register 0x5F

Bits	Prop	Description
0-2	WO brst BbClk	**Go2BitsyncThreshold**: *Threshold for Entering Bit Synchronization* Sets the required number of consecutive chirp pulses detected in the unsynchronized state to go to bit synchronization state. Default: 0x2 This register must be defined before: RxCmdStart = True
4-6	WO brst BbClk	**LeaveBitsyncThreshold**: *Threshold for Leaving Bit Synchronization* Sets the required number of consecutive non-detected chirp pulses (lost pulses) in bit synchronization state to return to the unsynchronized state. Default: 0x6 This register must be defined before: RxCmdStart = True

26.2.82 0x60 – Real-time clock TimeB transmission delay adjustment

Used for adjusting the RTC values before transmission due to hardware delays inside the transmitter.

Offset	RW	7	6	5	4	3	2	1	0
0x60	R								
	init	0	0	0	0	0	0	0	0
	W	RtcTimeBTxAdj							
	init	0	0	0	0	0	0	0	0

Table 138. Field properties for register 0x60

Bits	Prop	Description
0-7	WO brst BbClk	**RtcTimeBTxAdj**: *real-time clock TimeB Transmission Delay Adjustment* Before transmitting, the real-time clock control value is adjusted to compensate for transmitter delays. – NA_MacRtcTimeBTxAdj1M2Ary_IC_C (0x05) Adjustment for 1 Msymbols 2Ary. – NA_MacRtcTimeBTxAdj1M2AryFec_IC_C (0x07) Adjustment for 1 Msymbols 2Ary FEC. – NA_MacRtcTimeBTxAdj500k2Ary_IC_C (0x09) Adjustment for 500 ksymbols 2Ary. – NA_MacRtcTimeBTxAdj500k2AryFec_IC_C (0x0F) Adjustment for 500 ksymbols 2Ary FEC. Note: This value is added to the RTC value after reading due to the delay in the transmitter while shifting out this value.

26.2.83 0x61 – Real-time clock TimeB reception delay adjustment

Used for adjusting the RTC values after reception due to hardware delays inside the receiver.

Offset	RW	7	6	5	4	3	2	1	0
0x61	R								
	init	0	0	0	0	0	0	0	0
	W	RtcTimeBRxAdj							
	init	0	0	0	0	0	0	0	0

Table 139. Field properties for register 0x61

Bits	Prop	Description
0-7	WO brst BbClk	**RtcTimeBRxAdj**: *Real-time clock TimeB Reception Delay Adjustment* Adjusts the real-time clock value after reception to compensate for receiver delays. – NA_MacRtcTimeBRxAdj1M2Ary_IC_C (0x04) Adjustment for 1 Msymbols 2Ary. – NA_MacRtcTimeBRxAdj1M2AryFec_IC_C (0x09) Adjustment for 1 Msymbols 2Ary FEC. – NA_MacRtcTimeBRxAdj500k2Ary_IC_C (0x08) Adjustment for 500 ksymbols 2Ary. – NA_MacRtcTimeBRxAdj500k2AryFec_IC_C (0x0C) Adjustment for 500 ksymbols 2Ary FEC. Note: This value is added to the RTC value after reading due to delay in the transmitter while shifting out this value.

26.2.84 0x62 – Real-time clock

Used for setting the value of the real-time clock and for enabling Auto mode for TimeB packets.

Offset	RW	7	6	5	4	3	2	1	0
0x62	R								
	init	0	0	0	0	0	0	0	0
	W	Internal Use Only			RtcTimeB AutoMode			Rtc CmdRd	Rtc CmdWr
	init	0	0	0	0	0	0	0	0

Table 140. Field properties for register 0x62

Bits	Prop	Description
0	WO brst strb BbClk	**RtcCmdWr**: *Set RamRtcReg Value in Real-Time Clock* After writing the desired RTC value to RamRtcReg, this field transfers the value into the RTC circuit. This requires 62 µs after the command has been executed before the RTC is updated with the value from RamRtcReg. 0: Ignore 1: Set Value Note: The commands must not be used after a transmission has been initiated and while the transmit cycle has not ended. Furthermore, this command must not be used after a reception has been started and while the receive cycle has not ended.
1	WO brst strb BbClk	**RtcCmdRd**: *Set Real-Time Clock Value in RtcReg* Gets the value of the internal real-time clock and writes it into RamRtcReg, which is the RAM real-time clock buffer. This requires 31 µs after the command has been executed before RamRtcReg is updated with the RTC value. 0: Ignore 1: Set Value Note:The commands must not be used after a transmission has been initiated and while the transmit cycle has not ended. Furthermore, this command must not be used after a reception has been started and while the receive cycle has not ended.
4	WO brst BbClk	**RtcTimeBAutoMode**: *Real-Time Clock Auto Mode for TimeB Packets* Automatically sets the RTC values in TimeB packets for transmission and automatically stores the RTC values of received TimeB packets in the RTC. Otherwise, the RTC values in TimeB packets for transmission are taken from RAM in RamRtcTx and the RTC values in received TimeB packets is stored in RamRtcRx. – NA_RtcTimeBAutoModeOff_BC_C (0x0) The value of RamRtcTx is transmitted in the RTC field of the TimeB packet, while the RTC value in a received TimeB packet is stored in RamRtcRx. – NA_RtcTimeBAutoModeOn_BC_C (0x1) The RTC value for a transmitted TimeB packet is taken from the real-time clock, while the RTC value from a received TimeB is automatically stored in the real-time clock.

26.2.85 0x63 – AGC amplitude

Used in the alternative AGC to change gain value of the desired amplitude.

Offset	RW	7	6	5	4	3	2	1	0
0x63	R								
	init	0	0	0	0	0	0	0	0
	W				AgcAmplitude				
	init	0	0	0	0	1	1	0	0

Table 141. Field properties for register 0x63

Bits	Prop	Description
0-4	WO	**AgcAmplitude**: *AGC Amplitude* Select desired Amplitude for alternative AGC.

26.2.86 0x64 – Alternative AGC and AGC range offset

Used for switching between standard AGC and alternative AGC and for setting the value in AgcRangeOffset.

Offset	RW	7	6	5	4	3	2	1	0
0x64	R								
	init	0	0	0	0	0	0	0	0
	W	Use Alternative Agc				AgcRangeOffset			
	init	0	0	0	0	1	0	1	0

Table 142. Field properties for register 0x64

Bits	Prop	Description
0-3	WO	**AgcRangeOffset**: *AGC Range Offset* Sets a value in AgcRangeOffset for alternative AGC so that the alternative AGC is temporarily halted when: | Mean Value - Middle | > AgcRangeOffset This field has no meaning for standard AGC (when UseAlternativeAgc =0). Default value: 0x0A
7	WO	**UseAlternativeAgc**: *Use Alternative AGC* Switches between the standard and alternative AGC. 0: Disable 1: Enable Note: Do not switch between AGCs when the receiver is running.

27 Baseband registers

The following tables describe the memory locations of the baseband RAM.

Note: All registers in baseband RAM are both readable and writable.

27.1 Baseband register mapping

27.1.1 Auto/duplex mode

Auto/duplex page 0

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
0x80	0x85	R	RamStaAddr0							
		W								
0x86	0x87	R								
		W								
0x88	0x8D	R	RamStaAddr1							
		W								
0x8E	0x8F	R								
		W								
0x90	0x95	R	RamTxDstAddr							
		W								
0x96	0x97	R								
		W								
0x98	0x98	R	RamTxLength							
		W								
0x99	0x99	R	RamTx LCh	RamTx SeqN	RamTx FragC	RamTxLength				
		W								
0x9A	0xA7	R								
		W								
0xA8	0xAD	R	RamRxDstAddr							
		W								
0xAE	0xAF	R								
		W								
0xB0	0xB5	R	RamRxSrcAddr							
		W								
0xB6	0xB7	R								
		W								
0xB8	0xB8	R	RamRxLength							
		W								
0xB9	0xB9	R	RamRxLCh	RamRxSeqN	RamRxFragC	RamRxLength				
		W								
0xBA	0xDF	R								
		W								

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
0xE0	0xE5	R	RamRtcTx							
		W								
0xE6	0xE7	R								
		W								
0xE8	0xED	R	RamRtcRx							
		W								
0xEE	0xEF	R								
		W								
0xF0	0xFF	R	RamRtcReg							
		W								

Auto/duplex page 1

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
Register Space		R								
		W								
0x180	0x18F	R	RamTxRxCryptKey							
		W								
0x190	0x1BF	R								
		W								
0x1C0	0x1CF	R	RamTxCryptClock							
		W								
0x1D0	0x1DF	R								
		W								
0x1E0	0x1EF	R	RamRxCryptClock							
		W								
0x1F0	0x1FF	R								
		W								

Auto/duplex page 2

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
Register Space		R								
		W								
0x280	0x2FF	R	RamRxBuffer							
		W								

Auto/duplex page 3

		MSB			Bit Number				LSB
Offset	RW	7	6	5	4	3	2	1	0
Register Space	R								
	W								

		MSB	Bit Number						LSB
Offset	RW	7	6	5	4	3	2	1	0
0x380	0x3FF	R	RamTxBuffer						
		W							

27.1.2 Auto/simplex mode

Auto/simplex page 0

Same as *Section : Auto/duplex page 0*.

Auto/simplex page 1

Same as *Section : Auto/duplex page 1*.

Auto/simplex pages 2 and 3

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
0x280	0x2FF	R	RamTxRxBuffer							
		W								
0x380	0x3FF	R	RamTxRxBuffer							
		W								

27.1.3 Transparent/duplex mode

Transparent/duplex pages 0 and 1

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
0x080	0x0FF	R	RamRxTransBuffer							
		W								
0x180	0x1FF	R	RamRxTransBuffer							
		W								

Transparent/duplex pages 2 and 3

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
0x280	0x2FF	R	RamTxTransBuffer							
		W								
0x380	0x3FF	R	RamTxTransBuffer							
		W								

27.1.4 Transparent/simplex mode

Transparent/simplex pages 0, 1, 2, and 3

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
0x080	0x0FF	R	RamTxRxTransBuffer							
		W								
0x180	0x1FF	R	RamTxRxTransBuffer							
		W								
0x280	0x2FF	R	RamTxRxTransBuffer							
		W								
0x380	0x3FF	R	RamTxRxTransBuffer							
		W								

27.2 Description of baseband registers

27.2.1 Auto/duplex [Pg 0] 0x80 to 0x85 – source station address 0

Used to buffer the 48 bit address of the source station based on the IEEE standard addressing convention. The first 24 bits correspond to the Organizationally Unique Identifier assigned by IEEE, while the second 24 bits are administered locally. This address is set in the MACFrame of a Data, Brdcast, TimeB, and Reg2S packet.

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
0x80	0x85	R	RamStaAddr0							
		W								

Table 143. Field properties for registers 0x80 to 0x85

Bits	Prop	Description
0-47	RW brst BbClk RAM	**RamStaAddr0**: *Station Address Based on IEEE Addressing Convention* Stores a 48 bit universally administered IEEE address of the source station. If this convention is used, then set the first two bits of this field as follows: – RamStaAddr0[0] = 0 – RamStaAddr0[1] = 0 Default: 0x0 Note: It is not mandatory to use this convention for addressing.

27.2.2 Auto/duplex [Pg 0] 0x88 to 0x8D – source station address 1

Used to buffer the 48 bit address of the source station based on a configuration locally administered by the assignee. This address is set in the MACFrame of a Data, Brdcast, TimeB, or Reg2S packet.

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
0x88	0x8D	R	RamStaAddr1							
		W								

Table 144. Field properties for registers 0x88 to 0x8D

Bits	Prop	Description
0-47	RW brst BbClk RAM	**RamStaAddr1**: *Station Address Based on Locally Administered Configuration* Stores the second address for the station, a 48 bit locally administered address of the source station. If this convention is used, then set the first two bits of this field as follows: – RamStaAddr1[0] = 0 – RamStaAddr1[1] = 1 Note: It is not mandatory to use this convention for addressing.

27.2.3 Auto/duplex [Pg 0] 0x90 to 0x95 – TX destination address

Used to buffer a 48 bit address of the destination station to be set in the MACFrame of a Data or Brdcast packet that is to be transmitted. This destination address is used in all packet types, except the TimeB packet. In TimeB packets, this RAM location contains a real-time clock value.

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
0x90	0x95	R	RamTxDstAddr							
		W								

Table 145. Field properties for registers 0x90 to 0x95

Bits	Prop	Description
0-47	RW brst BbClk RAM	**RamTxDstAddr**: *Transmit Destination Address* Stores the 48 bit address of the destination station for a Data packet that is to be transmitted, as well as the 48 bit broadcast/ multicast address of a Brdcast packet to be transmitted. This register must be defined before: TxCmdStart = True. Usage: Auto mode only.

27.2.4 Auto/duplex [Pg 0] 0x98 to 0x99 – TX data length and bits for sw usage

Used to set the length of the data field in packets to be transmitted, as well as to set three bits that are available for use by software.

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
0x98	0x98	R	RamTxLength							
		W								
0x99	0x99	R	RamTxFragC TX Transparent Bits for SW usage			RamTxLength				
		W								

Table 146. Field properties for registers 0x98 to 0x99

Bits	Prop	Description
0-12	RW brst BbClk RAM	**RamTxLength**: *Data Field Length of a Packet to be Transmitted* Sets the number of bytes of the Data field of Data, Brdcast, and TimeB packets. This field will also be transmitted in a Req2s packet when using 3-way handshaking. This register must be defined before: TxCmdStart = True. Usage: Auto mode only.
13-15	RW brst BbClk RAM	**RamTxFragC** These bits are transparent for the TN100. There are only loaded from RAM together with the TxLength field and sent in the MACFrame. On the receiver side they can be read from the corresponding RX field called “RX Transparent Bits for SW usage.” In several applications these bits were used on software level to implement functionalities such as Link Control Channel, a sequential numbering scheme, or a packet fragmentation scheme. *Section 27.2.7: Auto/duplex [Pg 0] 0xB8 to 0xB9 – RX data length and bits for SW usage*. Usage: Auto mode only.

27.2.5 Auto/duplex [Pg 0] 0xA8 to 0xAD – RX destination address

Used to buffer the 48 bit address of the destination address field provided in the MACFrame of a received Data or Brdcast packet.

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
0xA8	0xAD	R	RamRxDstAddr							
		W								

Table 147. Field properties for registers 0xA8 to AD

Bits	Prop	Description
0-47	RW brst BbClk RAM	**RamRxDstAddr**: *Receive Destination Address* Stores the destination address from received Data and Brdcast packets. It allows for the retrieval of the destination address when in promiscuous mode in order to compare broadcast/multicast addresses (Brdcast packets). Valid when: – RxHeaderEnd event is triggered – RxCrc1 == RxCrc1Succ – RxPacketType == RxPacketTypeData | RxPacketTypeBrdcast Usage: Auto mode only.

27.2.6 Auto/duplex [Pg 0] 0xB0 to 0xB5 – RX source address

Used to buffer the 48 bit address of the source address field provided in the MACFrame of a received Data or Brdcast packet.

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
0xB0	0xB5	R	RamRxSrcAddr							
		W								

Table 148. Field properties for registers 0xB0 to 0xB5

Bits	Prop	Description
0-7	RW brst BbClk RAM	**RamRxSrcAddr**: *Receive Source Address* Stores the source address of the received Data or Brdcast packet. Valid when: – RxHeaderEnd event is triggered – RxCrc1 == RxCrc1Succ Usage: Auto mode only.

27.2.7 Auto/duplex [Pg 0] 0xB8 to 0xB9 – RX data length and bits for SW usage

Used to store a value which indicates the length of the data field in received packets, as well as to store three bits that are available for use by software.

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
0xB8	0xB8	R	RamRxLength							
		W								
0xB9	0xB9	R	RamRxFragC RX Transparent Bits for SW usage			RamRxLength				
		W								

Table 149. Field properties for registers 0xB8 to 0xB9

Bits	Prop	Description
0-12	RW brst BbClk RAM	**RamRxLength**: *Data Field Length of Received Packet* Stores the length (number of bytes) of the data field of Data, Brdcast, TimeB, Req2S, and Clr2S packets. Valid when: – RxHeaderEnd event is triggered – RxCrc1 == RxCrc1Succ Usage: Auto mode only
13-15	RW brst BbClk RAM	**RamRxFragC** These bits are transparent for the TN100. They are only stored to RAM together with the RxLength field received in the MACFrame. In several applications these bits were used on software level to implement functionalities such as Link Control Channel, a sequential numbering scheme, or a packet fragmentation scheme. Usage: Auto mode only.

27.2.8 Auto/duplex [Pg 0] 0xE0 to 0xE5 – TX Real-time clock buffer

Used to temporarily buffer the 48-bit value of the real-time clock to prepare time beacon (TimeB) packets for transmission.

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
0xE0	0xE5	R	RamRtcTx							
		W								

Table 150. Field properties for registers 0xE0 to 0xE5

Bits	Prop	Description
0-47	RW brst BbClk	**RamRtcTx**: *TX Real-Time Clock Value Buffer* Stores the value of the real-time clock for use by TimeB packets to be transmitted.

27.2.9 Auto/duplex [Pg 0] 0xE8 to 0xED – RX Real-Time Clock buffer

Used to temporarily buffer the value of the real-time clock from a received time beacon packet (TimeB).

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
0xE8	0xED	R	RamRtcRx							
		W								

Table 151. Field properties for registers 0xE8 to 0xED

Bits	Prop	Description
0-47	RW brst BbClk	**RamRtcRx**: *RX Real-Time Clock Value Buffer* Stores the value of the real-time clock that is provided in the MACFrame of a received TimeB packet.

27.2.10 Auto/duplex [Pg 0] 0xF0 to 0xFF – RX Real-Time Clock buffer

Used to buffer the 48-bit value of the internal real-time clock for use by software to read or write the RTC value.

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
0xF0	0xFF	R	RamRtcReg							
		W								

Table 152. Field properties for registers 0xF0 to 0xFF

Bits	Prop	Description
0-47	RW brst BbClk	**RtcReg**: *RAM Real-Time Clock Register* Buffer register for reading or writing the 48 bit internal real-time clock value. Since the real-time clock is updated every 30.517578125 µs, it is not directly accessible by the user. Note: RtcCmdRd reads the value of the real-time clock to this buffer. RtcCmdWr writes a value in this buffer to the real-time clock.

27.2.11 Auto/duplex [Pg 1] 0x100 to 0x17F – register (mirrored)

The register (0x00 to 0x7F) is mirrored to RAM locations 0x100 to 0x17F.

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
0x100	0x17F	R	Mirrored Register (0x00 to 0x7F)							
		W								

27.2.12 Auto/duplex [Pg 1] 0x180 to 0x1BF – encryption key

Used to store the four 128 bit encryption keys used for cipher generation.

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
0x180	0x18F	R	RamTxRxCryptKey[0]							
		W								
0x190	0x19F	R	RamTxRxCryptKey[1]							
		W								
0x1A0	0x1AF	R	RamTxRxCryptKey[2]							
		W								
0x1B0	0x1BF	R	RamTxRxCryptKey[3]							
		W								

Table 153. Field properties for registers 0x100 to 0x17F

Bits	Prop	Description
Per key:0-127 Total:0-511	RW brst BbClk	**RamTxRxCryptKey**: *Encryption keys* Stores the four 128 bit encryption keys used as inputs of the cipher stream generator. Value: RamTxRxCryptKey[CryptId], CryptId = 0...3 This register must be defined before: TxCmdStart = True or RxCmdStart = True. Usage: Auto mode only.

27.2.13 Auto/duplex [Pg 1] 0x1C0 to 0x1CF – TX encryption clock value

Used to store the four 32-bit clock values used for cipher generation (encryption).

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
0x1C0	0x1C3	R	RamTxCryptClock[0]							
		W								
0x1C4	0x1C7	R	RamTxCryptClock[1]							
		W								
0x1C8	0x1CB	R	RamTxCryptClock[2]							
		W								
0x1CC	0x1CF	R	RamTxCryptClock[3]							
		W								

Table 154. Field properties for registers 0x1C0 to 0x1CF

Bits	Prop	Description
Per value:0-31 Total: 0-127	RW brst BbClk RAM	**RamTxCryptClock**: *Encryption Clock Value* Stores the four publicly known 32 bit clock values used as inputs of the cipher stream generator for encryption. Value: RamTxCryptClock[CryptId], CryptId = 0…3. This register must be defined before: TxCmdStart = True. Usage: Auto mode only.

27.2.14 Auto/duplex [Pg 1] 0x1E0 to 0x1EF – RX decryption clock value

Used to store the four 32 bit clock values used as input for cipher generation (decryption).

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
0x1E0	0x1E3	R W	RamRxCryptClock[0]							
0x1E4	0x1E7	R W	RamRxCryptClock[1]							
0x1E8	0x1EB	R W	RamRxCryptClock[2]							
0x1EC	0x1EF	R W	RamRxCryptClock[3]							

Table 155. Field properties for registers 0x1E0 to 0x1EF

Bits	Prop	Description
Per value:0-31 Total:0-127	RW brst BbClk RAM	**RamRxCryptClock**: *Receive Decryption Clock* Stores the four publicly known 32 bit clock values used as inputs of the cipher stream generator for decryption. Value: RxCryptClock[CryptId], CryptId = 0…3. This register must be defined before: RxCmdStart = True.

27.2.15 Auto/duplex [Pg 2] 0x200 to 0x27F – register (mirrored)

The chip register (0x00 to 0x7F) is mirrored to RAM locations 0x200 to 0x27F.

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
0x200	0x27F	R W	Mirrored Register (0x00 to 0x7F)							

27.2.16 Auto/duplex [Pg 2] 0x280 to 0x2FF – RX data buffer

Used to store 128 bytes (1024 bits) of data in two buffers of 64 bytes each from a received Data packet.

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
0x280	0x2FF	R	RamRxBuffer							
		W								

Table 156. Field properties for registers 0x280 to 0x2FF

Bits	Prop	Description
0-1023	RW brst BbClk RAM	**RamRxBuffer**: *128 Byte RX Data Buffer* Stores in two buffers of 64 bytes each the received data from Data packets. Note: Data in buffer are valid when the RxBufferRdy event is triggered.

27.2.17 Auto/duplex [Pg 3] 0x300 to 0x37F – register (mirrored)

The register (0x00 to 0x7F) is mirrored to RAM locations 0x300 to 0x37F.

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
0x200	0x27F	R	Mirrored Register (0x00 to 0x7F)							
		W								

27.2.18 Auto/duplex [Pg 3] 0x380 to 0x3FF – TX data buffer

Used to store 128 bytes (1024 bits) of data in two buffers of 64 bytes each for a Data packet to be transmitted.

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
0x380	0x3FF	R	RamTxBuffer							
		W								

Table 157. Field properties for registers 0x380 to 0x3FF

Bits	Prop	Description
0-1023	RW brst BbClk RAM	**RamTxBuffer**: *128 Byte TX Data Buffer* Stores in two buffers of 64 bytes each the data to be set in of a Data packet that is to be transmitted.

27.2.19 Auto/simplex [Pg 0] 0x80/0x0FF – same as auto/duplex [Pg 0]

Auto/Simplex Page 0 is identical to Auto/Duplex Page 0, which is replicated below.

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
Register Space		R	Mirrored Register (0x00 to 0x7F)							
		W								
0x80	0x85	R	RamStaAddr0							
		W								

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
0x86	0x87	R	Reserved							
		W								
0x88	0x8D	R	RamStaAddr1							
		W								
0x8E	0x8F	R	Reserved							
		W								
0x90	0x95	R	RamTxDstAddr							
		W								
0x96	0x97	R	Reserved							
		W								
0x98	0x98	R	RamTxLength							
		W								
0x99	0x99	R	TX Transparent Bits for SW usage			RamTxLength				
		W								
0x9A	0xA7	R	Reserved							
		W								
0xA8	0xAD	R	RamRxDstAddr							
		W								
0xAE	0xAF	R	Reserved							
		W								
0xB0	0xB5	R	RamRxSrcAddr							
		W								
0xB6	0xB7	R	Reserved							
		W								
0xB8	0xB8	R	RamRxLength							
		W								
0xB9	0xB9	R	RX Transparent Bits for SW usage			RamRxLength				
		W								
0xBA	0xDF	R	Reserved							
		W								
0xE0	0xE5	R	RamRtcTx							
		W								
0xE6	0xE7	R	Reserved							
		W								
0xE8	0xED	R	RamRtcRx							
		W								
0xEE	0xEF	R	Reserved							
		W								
0xF0	0xFF	R	RamRtcReg							
		W								

27.2.20 Auto/simplex [Pg 1] 0x180/0xEFF – same as auto/duplex [Pg 1]

Auto/Simplex Page 1 is identical to Auto/Duplex Page 1, which is replicated below.

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
Register Space		R	Mirrored Register (0x00 to 0x7F)							
		W								
0x180	0x18F	R	RamTxRxCryptKey							
		W								
0x190	0x1BF	R	Reserved							
		W								
0x1C0	0x1CF	R	RamTxCryptClock							
		W								
0x1D0	0x1DF	R	Reserved							
		W								
0x1E0	0x1EF	R	RamRxCryptClock							
		W								
0x1F0	0x1FF	R	Reserved							
		W								

27.2.21 Auto/simplex [Pg 2/3] 0x280 to 0x3FF – TX/RX data buffer

Used for storing sequentially transmit and receive data, where the entire 256 bytes are divided into two 128 byte (2048 bit) buffers as a combined receive/transmit buffer. As this is Simplex mode, the transmit and receive buffer operations are dependent on each other, in that the receive operation must be completed before the transmit operation can be initiated. Consequently, the buffer for the RX data and for the TX data is double the size, in this case, 256 bytes.

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
Register Space		R	Mirrored Register (0x00 to 0x7F)							
		W								
0x280	0x2FF	R	RamTxRxBuffer							
		W								
Register Space		R	Mirrored Register (0x00 to 0x7F)							
		W								
0x380	0x3FF	R	RamTxRxBuffer							
		W								

Table 158. Field properties for registers 0x280 to 0x3FF

Bits	Prop	Description
0-2047	RW brst BbClk RAM	**RamTxRxBuffer**: *256 Byte Shared Transmit/Receive Data Buffer* Stores both received data and data for transmission (sequentially) in a combined RX/TX buffer.

27.2.22 Transparent/duplex page [Pg 0/1] 0x080 to 0x1FF – RX data buffer

The RX data buffer, which is used for storing the MACFrame from a received packet, is a 256 byte buffer divided into two 128 byte buffers.

As this is Transparent mode, software is responsible fro maintaining MACFrame variables. As this is Duplex mode, received data can saved to this buffer while data to be transmitted can be simultaneously written to the TX data buffer `RamTxTransBuffer`.

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
Register Space		R	Mirrored Register (0x00 to 0x7F)							
		W								
0x080	0x0FF	R	RamRxTransBuffer							
		W								
Register Space		R	Mirrored Register (0x00 to 0x7F)							
		W								
0x180	0x1FF	R	RamRxTransBuffer							
		W								

Table 159. Field properties for registers 0x080 to 0x1FF

Bits	Prop	Description
0-2047	RW brst BbClk RAM	**RamRxTransBuffer**: *256 Byte Receive Buffer* Stores the received data in two 128 byte buffers.

27.2.23 Transparent/duplex page [Pg 2/3] 0x280 to 0x3FF – TX data buffer

The TX Data Buffer, which is used for storing completed MACFrames from software for transmission, is a 256-byte buffer divided into two 128-byte buffers.

As this is Transparent mode, software is responsible fro maintaining MACFrame variables. As this is Duplex mode, data for transmission can written to this buffer while received data can be simultaneously saved to the RX Data Buffer `RamRxTransBuffer`.

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
Register Space		R	Mirrored Register (0x00 to 0x7F)							
		W								
0x280	0x2FF	R	RamTxTransBuffer							
		W								
Register Space		R	Mirrored Register (0x00 to 0x7F)							
		W								
0x380	0x3FF	R	RamTxTransBuffer							
		W								

Table 160. Field properties for registers 0x280 to 0x3FF

Bits	Prop	Description
0-2047	RW brst BbClk RAM	**RamTxTransBuffer**: *256 Byte Transmit Buffer* Stores the data to be transmitted in two 128 byte buffers.

27.2.24 Transparent/simplex [Pg 0/1/2/3] 0x080/0x3FF – RX/TX data buffer

The RX/TX data buffer is a single 512-byte buffer divided into four 128-byte buffers. It is used for storing sequentially MACFrames from a received packet to be read from software, or MACFrames provided by software to be transmitted in packets over the air.

As this is Transparent mode, software is responsible fro maintaining MACFrame variables. As this is Simplex mode, the transmit and receive buffer operations are dependent of each other so that a receive operation must be completed first before a transmit operation can be initiated. Consequently, the buffer size available for the RX or TX data is the complete baseband RAM memory space of 512 bytes.

Offset			MSB			Bit Number				LSB
From	To	RW	7	6	5	4	3	2	1	0
Register Space		R	Register (0x00 to 0x7F)							
		W								
0x080	0x0FF	R	RamTxRxTransBuffer							
		W								
Register Space		R	Mirrored Register (0x00 to 0x7F)							
		W								
0x180	0x1FF	R	RamTxRxTransBuffer							
		W								
Register Space		R	Mirrored Register (0x00 to 0x7F)							
		W								
0x280	0x2FF	R	RamTxRxTransBuffer							
		W								
Register Space		R	Mirrored Register (0x00 to 0x7F)							
		W								
0x380	0x3FF	R	RamTxRxTransBuffer							
		W								

Table 161. Field properties for registers 0x080 to 0x3FF

Bits	Prop	Description
0-4095	RW brst BbClk RAM	**RamTxRxTransBuffer**: *512 byte Shared Transmit/Receive Buffer* Sequentially stores the received data or the data to be transmitted in a combined 512 byte RX/TX buffer.

28 Ordering information

Table 162. Ordering information

Part number	Temperature range	Package	Packing	Marking
TN100Q	-40°C to +85°C	ECOPACK® QFN48 Package	Tube	TN100Q
TN100QT	-40°C to +85°C	ECOPACK® QFN48 Package	Tape & Reel	TN100QT

29 Electrical characteristics

29.1 Absolute maximum ratings

Table 163. Absolute maximum ratings

Parameter	Value[1]	Unit
Temperature		
Maximum operating temperature	85	°C
Maximum junction temperature	95	°C
Maximum storage temperature	125	°C
Reflow solder temperature (lead-free package)	260	°C
Voltages		
Power supply voltage VDDA (analogue block)	2.7	V
Power supply voltage VDDD (digital block)	2.7	V
Power		
Total power dissipation FDMA-CSS mode	130	mW
Total power dissipation CSS Ranging mode	172	mW
Electrostatic Discharge Protection (ESD Protection)		
Maximum ESD input potential, Human Body Model	1000	V

1. It is critical that the ratings provided in Absolute Maximum Ratings be carefully observed. Stress exceeding one or more of these limiting values may cause permanent damage to the device.

29.2 Nominal conditions

The nominal conditions are as specified below, unless specified otherwise.

- Reference design used.
- T_{junct} = 30°C
- $V_{SSA} = V_{SSD}$ = GND
- $V_{DDA} = V_{DDD}$ = +2.5 V
- Transmission / reception at 250 kbps
- Nominal frequency bandwidth of the channel B = 22 MHz at -30 dBr
- Raw data mode
- No CRC
- No FEC
- No encryption
- Bit scrambling
- No ranging
- BER = 0.001 during receive mode
- RF output power (PEP) during transmit phase = 0 dBm EIRP measured during continuous transmission

- Nominal process
- All RF ports are impedance matched according to the specification.
- All RF power are measured on the IC terminals (pins)
- For link distance measurement, two identical TN100 systems are used

Table 164. Electrical data

Parameter	Value	Unit
Maximum supply voltage	2.7	V
Minimum supply voltage	2.3	V
Maximum output power	0	dBm
Maximum data rate	2	Mbps
Sensitivity at nominal conditions	-95	dBm
Sensitivity at nominal conditions and FEC on	-97	dBm
Supply current		
In transmit mode at -10 dBm output power & nominal conditions	25	mA
In transmit mode at 0 dBm output power & nominal conditions	30	mA
In receive mode & nominal conditions	33	mA
In shut-down mode	2	µA
Operating temperature range	-40 to +85	°C
Frequency channels (FDMA mode, non-overlapping channels)		
Number of frequency channels	3	Number
Center frequency of channel no. 1 (Europe)	2412	MHz
Center frequency of channel no. 2 (Europe)	2442	MHz
Center frequency of channel no. 3 (Europe)	2472	MHz
Center frequency of channel no. 1 (USA)	2412	MHz
Center frequency of channel no. 2 (USA)	2437	MHz
Center frequency of channel no. 3 (USA)	2462	MHz
Frequency channels (FDMA mode, overlapping channels)		
Number of frequency channels	7	Number
Center frequency of channel no. 1 (overlapping)	2412	MHz
Center frequency of channel no. 2 (overlapping)	2422	MHz
Center frequency of channel no. 3 (overlapping)	2432	MHz
Center frequency of channel no. 4 (overlapping)	2442	MHz
Center frequency of channel no. 5 (overlapping)	2452	MHz
Center frequency of channel no. 6 (overlapping)	2462	MHz
Center frequency of channel no. 7 (overlapping)	2472	MHz
Nominal frequency bandwidth of the channel at -30 dBr	22	MHz
Nominal frequency bandwidth for ranging at -30 dBr	83	MHz
Typical operational voltages		
Typical Power supply voltage V_{DDA} (analogue block)	2.5	V
Typical Power supply voltage V_{DDD} (digital block)	2.5	V

29.3 General DC parameters

Table 165. General DC parameters

Parameter	Value	Unit
Operating frequency range	2.4 GHz ISM Band	–
Supply voltage range	2.3 … 2.7	V
Modulation method	Chirp	–
Operating temperature range	-40 … +85	°C
Supply current for individual blocks		
Tx block (P_{out} = 0 dBm)	23	mA
Tx block, ranging with increased accuracy (P_{out} = 0 dBm)	30	mA
Rx block	27	mA
Rx block, ranging with increased accuracy	34	mA
Digital part, Tx mode	7	mA
Digital part, Tx mode, ranging with increased accuracy	9	mA
Digital part, Rx mode	6	mA
Digital part, Rx mode, ranging with increased accuracy	10	mA
Total supply current		
Tx mode (P_{out} = 0 dBm)	30	mA
Tx mode, ranging with increased accuracy (P_{out} = 0 dBm)	39	mA
Rx mode	33	mA
Rx mode, ranging with increased accuracy	44	mA
VDD1V2_Cap (Pin 28): 1.2 V digital power supply decoupling		
Decoupling capacitance (typical)	100	nF
VBalun (Pin 47): DC voltage for RF output stage. This must be fed to TxN and TxP using bias-Tees or a balun transform with center tap. Min and max Values for decoupling bypass capacitor are shown below. (It is not a block capacitor).Also refer to the reference design.		
Decoupling bypass capacitor Min	27	pF
Decoupling bypass capacitor Max	47	pF
µCVcc (pin 29): switchable power supply for external microcontroller		
Maximum capacitive load	10	µF
Maximum output current	10	mA
RRef (pin 2): external precise reference resistor		
Resistance	10	kΩ
Recommended resistance tolerance	1	%

29.4 Transmitter (TX)

29.4.1 General parameters

Table 166. Transmitter – general parameters

Parameter	Value	Unit
Transmitter nominal output power	0	dBm
Dynamic for output power control	≥ 33	dB
Number of steps for output power control	64	Number
Load impedance	200	Ohm
Type of load	Balanced	–
Transmitter spurious outputs (1 GHz ... 12.5 GHz)	-80	dBm/Hz
Transmitter carrier suppression	-20	dBc
Number of frequency channels (FDMA Mode, non-overlapping channels), according to IEEE 802.15.4a standard	3	Number
Centre frequency of FDMA channel no. 1 (Europe)	2412	MHz
Center frequency of FDMA channel no. 2 (Europe)	2442	MHz
Centre frequency of FDMA channel no. 3 (Europe)	2472	MHz
Centre frequency of FDMA channel no. 1 (USA)	2412	MHz
Centre frequency of FDMA channel no. 2 (USA)	2437	MHz
Centre frequency of FDMA channel no. 3 (USA)	2462	MHz
Number of frequency channels (FDMA mode, overlapping channels), according to IEEE 802.15.4a standard	7	Number
Center frequency of channel no. 1 (overlapping)	2412	MHz
Center frequency of channel no. 2 (overlapping)	2422	MHz
Center frequency of channel no. 3 (overlapping)	2432	MHz
Center frequency of channel no. 4 (overlapping)	2442	MHz
Center frequency of channel no. 5 (overlapping)	2452	MHz
Center frequency of channel no. 6 (overlapping)	2462	MHz
Center frequency of channel no. 7 (overlapping)	2472	MHz
Carrier frequency accuracy (relative), CSS mode	± 70	ppm
Carrier frequency accuracy (absolute), CSS mode	± 171	ppm
VBalun (Pin 47): DC voltage for RF output stage. This must be fed to TxN and TxP using bias-Tees or a balun / transformer with center tap. Min and max Values for decoupling bypass capacitor are shown below. RF ceramic type with low serial inductance is recommended. See also the reference design.		
Decoupling bypass capacitor min capacitance	27	pF
Decoupling bypass capacitor max capacitance	47	pF

29.4.2 Programmable RAM for chirp sequencer

Table 167. Transmitter - programmable RAM for chirp sequencer

Parameter	Value	Unit
Memory type	Single port RAM	–
Memory organization	24 x 256	Number
Number of memory banks	2	Number
Width of output data bus for each memory bank	6	Bit
Number of memory cells in one bank	256	Number
Width of address bus	8	Bit

29.4.3 Chirp specification (CSS)

Transmitter - chirp specification (CSS)

Parameter	Value	Unit
Chirp duration (programmable)	1, 2 and 4	µs
Symbol rate		
Nominal	1	Mbaud
Reduced	0.5 and 0.25	Mbaud
Chirp Sequencer Clock Frequency f_{Chirp}, FDMA-CSS mode	32	MHz
Chirp Sequencer Clock Frequency f_{Chirp}, CSS Ranging mode	244.175	MHz

29.5 Receiver (RX)

29.5.1 General parameters

Table 168. Receiver – general parameters

Parameter	Value	Unit
Receiver sensitivity for CSS, 250 kbps at BER = 10^{-3}, nominal conditions	-95	dBm
Receiver sensitivity for CSS, 250 kbps at BER = 10^{-3}, nominal conditions, except FEC ON	-97	dBm
Input impedance	200	Ohm
Type of RX input	Balanced	–
Typical Noise Figure	3.5	dB
Maximum Noise Figure	5	dB
Maximum input CSS signal at BER=10^{-3}	-20	dBm
Number of frequency channels (FDMA Mode, non-overlapping channels), according to IEEE 802.15.4a standard	3	Number

Table 168. Receiver – general parameters (continued)

Parameter	Value	Unit
Centre frequency for FDMA channel no. 1 (Europe)	2412	MHz
Centre frequency for FDMA channel no. 2 (Europe)	2442	MHz
Centre frequency for FDMA channel no. 3 (Europe)	2472	MHz
Centre frequency for FDMA channel no. 1 (USA)	2412	MHz
Centre frequency for FDMA channel no. 2 (USA)	2437	MHz
Centre frequency for FDMA channel no. 3 (USA)	2462	MHz
Number of frequency channels (FDMA Mode, overlapping channels), according to IEEE 802.15.4a standard	7	Number
Center frequency of channel no. 1 (overlapping)	2412	MHz
Center frequency of channel no. 2 (overlapping)	2422	MHz
Center frequency of channel no. 3 (overlapping)	2432	MHz
Center frequency of channel no. 4 (overlapping)	2442	MHz
Center frequency of channel no. 5 (overlapping)	2452	MHz
Center frequency of channel no. 6 (overlapping)	2462	MHz
Center frequency of channel no. 7 (overlapping)	2472	MHz
Nominal frequency bandwidth of the channel @ -30 dB	22	MHz
LO frequency accuracy (relative), CSS mode	± 70	ppm
LO frequency accuracy (absolute), CSS mode	± 171	kHz

29.5.2 Radio signal strength indicator (RSSI)

Table 169. Receiver - RSSI

Parameter	Value	Unit
Absolute accuracy	± 6	dB
Resolution	± 2	dB

29.6 Dynamic performance

Table 170. Dynamic performance

Parameter	Value	Unit
RX initialization time	≤ 60	µs
TX initialization time	≤ 24	µs
Switch TX-RX, ACK to DATA mode	≤ 3	µs
Switch TX-RX, DATA to DATA mode	≤ 3	µs
Switch TX-RX, DATA to ACK mode	≤ 3	µs
Switch RX-TX, ACK to DATA mode	≤ 24	µs

Table 170. Dynamic performance (continued)

Parameter	Value	Unit
Switch RX-TX, DATA to DATA mode	≤ 24	µs
Switch RX-TX, DATA to ACK mode	≤ 8	µs
Start-up time for 32 MHz reference oscillator	≤ 5	ms
Start-up time for 32768 Hz RTC oscillator	≤ 2	s

29.7 Quartz controlled oscillator for reference frequency

Table 171. Quartz controlled oscillator for reference frequency

Parameter	Value	Unit
Frequency f_{REF}	32	MHz
Oscillation type of the reference quartz resonator	Fundamental	–
Recommended max. frequency temperature coefficient of the reference quartz resonator, CSS mode only	± 20	ppm
Recommended max. frequency tolerance of the reference quartz resonator, CSS mode only	± 10	ppm
Recommended max. aging of the reference quartz resonator in 10 years, CSS mode only	± 10	ppm
Accuracy of the reference quartz resonator, CSS and 15.4a compatible mode, including temperature coefficient, frequency tolerance aging, etc.	40	ppm
Max. equivalent serial resistance of the reference quartz resonator	40	Ohm
Recommended load capacitance	12	pF
Input for external signal with frequency f_{REF}	Yes	
Pin name for external signal with frequency f_{REFTC}	Xtal32MP	

29.8 Quartz-controlled oscillator for real-time clock (RTC)

Table 172. Quartz-controlled oscillator for real-time clock (RTC)

Parameter	Value	Unit
Frequency f_{RTC}	32768	Hz
Oscillation type of the RTC quartz resonator	Fundamental	
Recommended accuracy of the quartz resonator	± 20	ppm
Maximum equivalent serial resistance of the RTC quartz resonator	80	kOhm
Recommended load capacitance	12.5	pF
Input for external signal with frequency f_{RTC}	Yes	
Pin name for external signal with frequency f_{RTC}	Xtal32kP	

29.9 Local oscillator

Table 173. Local oscillator (LO)

Parameter	Value	Unit
Number of frequency channels (FDMA Mode, non-overlapping channels), according to IEEE 802.15.4a standard	3	Number
Nominal LO frequency f_{LO1E} for FDMA channel no. 1 (Europe)	2412	MHz
Nominal LO frequency f_{LO2E} for FDMA channel no. 2 (Europe)	2442	MHz
Nominal LO frequency f_{LO3E} for FDMA channel no. 3 (Europe)	2472	MHz
Nominal LO frequency f_{LO1U} for FDMA channel no. 1 (USA)	2412	MHz
Nominal LO frequency f_{LO2U} for FDMA channel no. 2 (USA)	2437	MHz
Nominal LO frequency f_{LO3U} for FDMA channel no. 3 (USA)	2462	MHz
Number of frequency channels (FDMA Mode, overlapping channels), according to IEEE 802.15.4a standard	7	Number
Center frequency of channel no. 1 (overlapping)	2412	MHz
Center frequency of channel no. 2 (overlapping)	2422	MHz
Center frequency of channel no. 3 (overlapping)	2432	MHz
Center frequency of channel no. 4 (overlapping)	2442	MHz
Center frequency of channel no. 5 (overlapping)	2452	MHz
Center frequency of channel no. 6 (overlapping)	2462	MHz
Center frequency of channel no. 7 (overlapping)	2472	MHz
Accuracy of the LO frequency, typical CSS mode	± 70	ppm
Accuracy of the LO frequency, worst case, CSS mode	± 100	ppm
Accuracy of the LO frequency, typical 15.4a mode	TBD	ppm
Accuracy of the LO frequency, worst case, 15.4a mode	40	ppm
Minimum LO frequency adjustment range	83.5	MHz
LO noise rejection	-40	dBm

29.10 Digital interface to sensor/actor

Table 174. Digital Interface to sensor / actor

Parameter	Value	Unit
Number of general-purpose inputs/outputs	4	Number
Width of each interface	1	Bit
Direction	In/out (bi-directional, open-drain with pull-up	–
Type	Programmable	–
Logic input capacitance	1.5	pF
Output high level	(0.8 … 1.0) x V_{DDD}	V
Output low level	(0 … 0.2) x V_{DDD}	V
Minimum value of input high level	(0.6 … 0.7) x V_{DDD}	V
Maximum value of input low level	(0.3 … 0.4) x V_{DDD}	V

29.11 Interface to digital controller

Table 175. Interface to digital controller

Pin Name	Parameter	Specification
µCIRQ	Push-pull	Power-down mode, leakage current (GND)(1) << 1 µA, slow
SpiTxD, SpiClk, SpiSSn	Input	LVC(2), power down mode, leakage current (GND)* << 1 µA, slow
SpiTxD	Push-pull	LVC(2), power down mode, leakage current (GND)* << 1 µA, slow

1. Sum of all leakage currents plus internal stand-by consumption must not exceed the total stand-by power consumption
2. LVC, V_{DDD} = 2.3 V: V_{OH} = 2.0 V, V_{IH} = 1.7 V, V_{OL} = 0.2 V, V_{IL} = 0.7 V. V_{DDD} = 2.3 … 2.7 V: V_{OH} = 2.4 V, V_{IH} = 1.7...2.0 V, V_{OL} = 0.2 V, V_{IL} = 0.8 V

29.12 Power supply for the external digital microcontroller

Table 176. Power supply for external digital microcontroller

Parameter	Value	Unit
Typical output voltage at I_{Load} = 10 mA	VDD-0.04 (1)	V
Maximum capacitive load at µCVcc	10	µF
Maximum output current	10	mA
Typical start-up time at I_{Load} = 10 mA, C_{Load} = 10 uF	1.5	ms

1. V_{DD} = 2.3 … 2.7 V

30 Package mechanical data

Figure 86. VFQFPN2-48 package outline

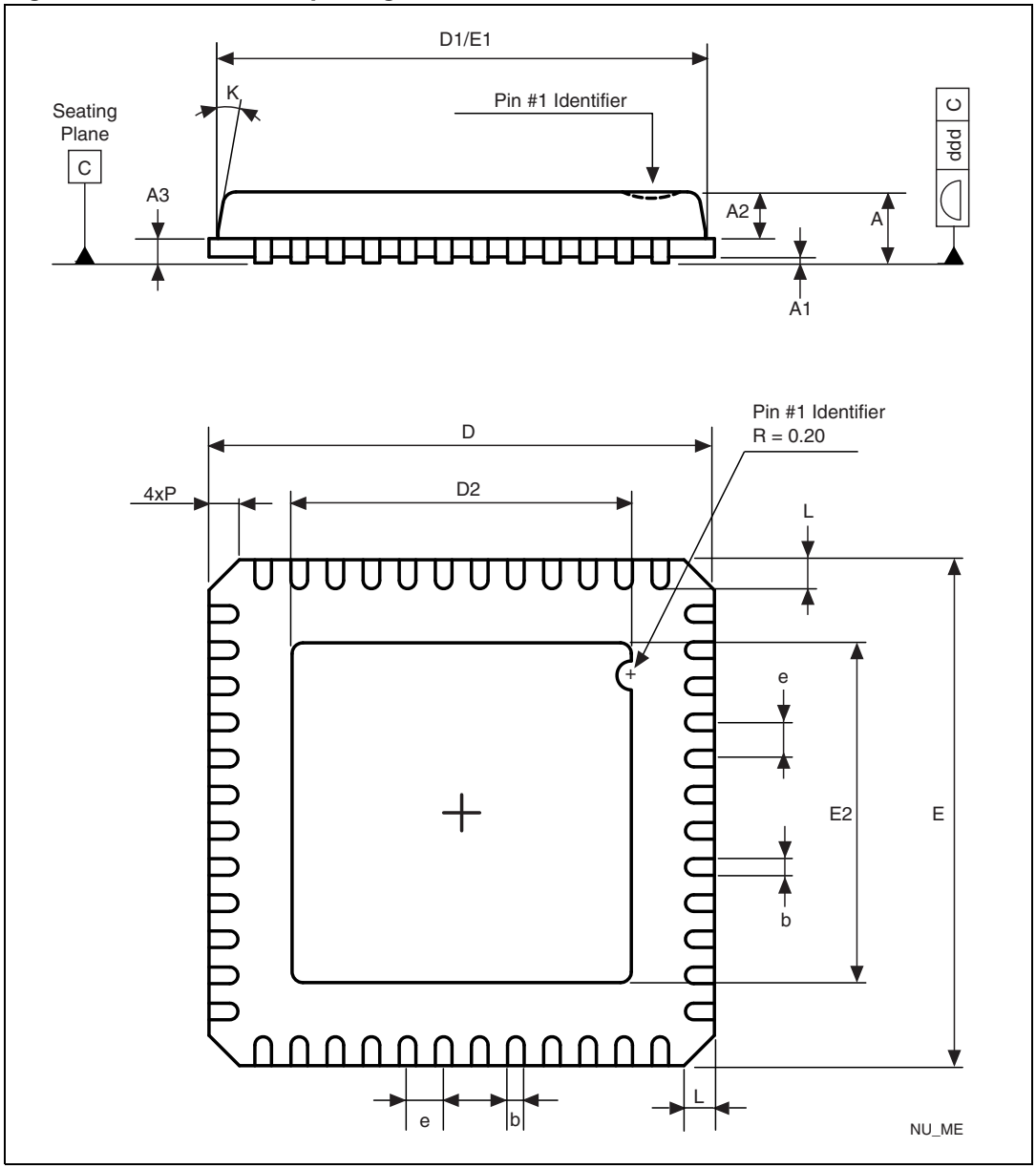


Table 177. VFQFPN2-48 package mechanical data

Symbol	millimeters			inches		
	Min	Typ	Max	Min	Typ	Max
A	0.800	0.900	1.000	0.0315	0.0354	0.0394
A1		0.020	0.050		0.0008	0.0020
A2		0.650	1.000		0.0256	0.0394
A3		0.200			0.0079	
b	0.180	0.250	0.300	0.0071	0.0098	0.0118
D	6.850	7.000	7.150	0.2697	0.2756	0.2815
D1		6.750			0.2657	
D2						
E	6.850	7.000	7.150	0.2697	0.2756	0.2815
E1		6.750			0.2657	
E2						
e		0.500			0.0197	
L	0.300	0.400	0.500	0.0118	0.0157	0.0197
P			0.600			0.0236
K	12°			12°		
ddd			0.080			0.0031

Appendix A Attributes and constants

The following table lists the attributes and constants referred to in this document.

Table 178. Attributes and constants

Symbol	Value	Description
MacAckTimeMax	MacSifsTime + 2 • MacAirDelTimeMax = 24 µs	Maximum time out for receiving an Ack packet.
MacAirDelTimeMax	< 8 µs (< 2.4 km)	MAC Air Delay Time Maximum (maximum air interface delay).(1)
MacArqCntMax	0 to 14	Maximum allowed number of retransmission attempts for Data packets.
MacBrdcastLengthMax	8192 bytes	Maximum allowed length of the data field of Brdcast packets in bytes.
MacCifsTime	MacSifsTime + 2 • MacAirDelTimeMax = 24 µs	Carrier Sense Interframe Space Time (CIFS).(2)
MacClr2STimeMax	MacSifsTime + 2 • MacAirDelTimeMax = 24 µs	Maximum time out for receiving a Clr2S packet.
MacDataLengthMax	8192 bytes	Maximum allowed length of the data field of Data packets in bytes.
MacPreambleSymbols	30	Number of preamble symbols.
MacSifsTime	8 µs	Switching Interframe Space Time (SIFS) – (switching time from reception to transmission of symbols (air interface)).(3)
MacSlotTime	MacSifsTime + 2 • MacAirDelTimeMax = 24 µs	Duration of an back-off slot.(4)
MacSlotUncertainty	MacAirDelTimeMax = 8 µs	Uncertainty of back-off slot alignment due to air propagation delay.(5)
MacTailSymbols	4	Number of tail symbols.

1. See *Figure 88. Carrier sense interframe space time (CIFS) on page 222* and *Figure 89. Air interface delay on page 223*.
2. See *Figure 88. Carrier sense interframe space time (CIFS) on page 222*.
3. See *Figure 87. Switching interframe space time (SIFS) on page 222* and *Figure 88. Carrier sense interframe space time (CIFS) on page 222*.
4. See *Figure 89. Air interface delay on page 223*.
5. See *Figure 89. Air interface delay on page 223*.

The following three figures illustrate the attributes and constants defined in this appendix.

Figure 87. Switching interframe space time (SIFS)

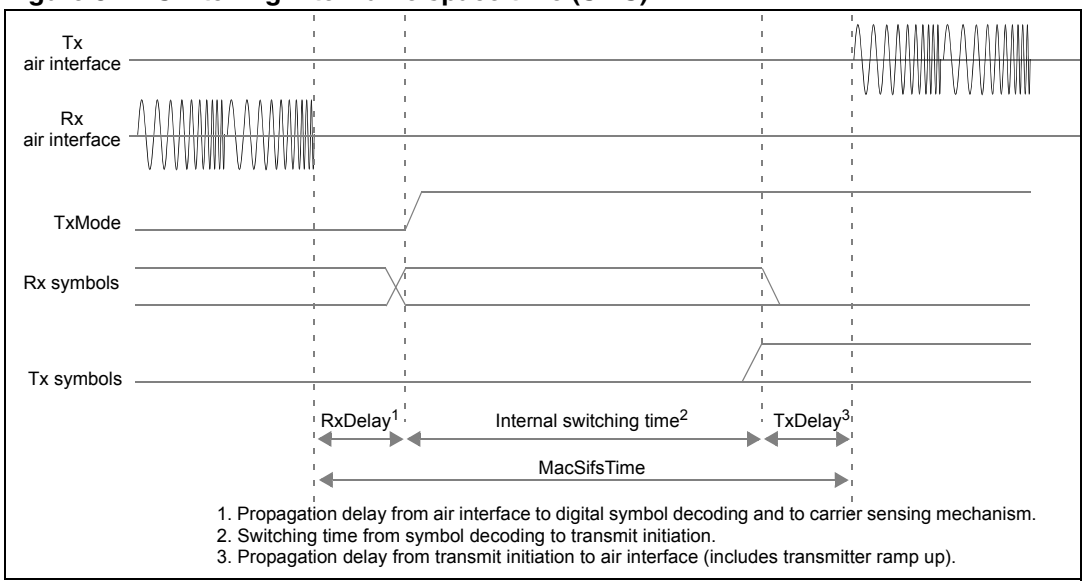


Figure 88. Carrier sense interframe space time (CIFS)

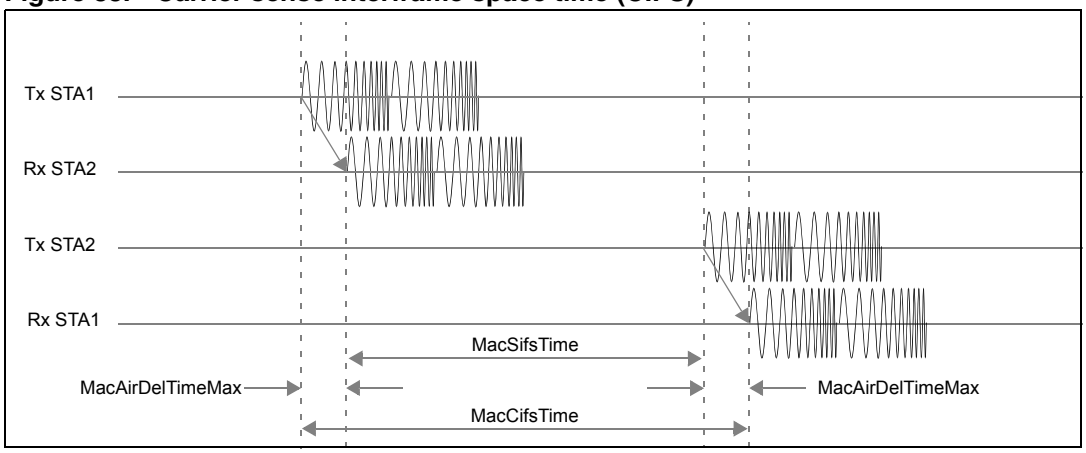
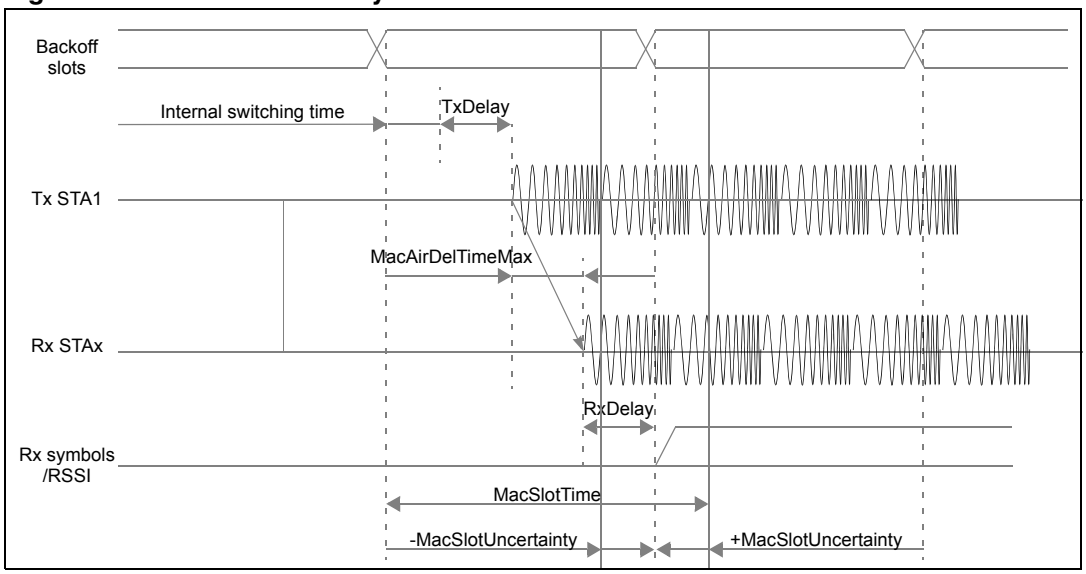


Figure 89. Air interface delay



Appendix B Default register settings

Table 179. List of default register settings

Register	Bit	RW	Field	Default	Description
0x00	0	RW	SpiBitOrder	0x0	LSB Bit Order
	1	RW	SpiTxDriver	0x0	Open-Drain
	2	RW	IrqPolarity	0x0	IRQ is low active
	3	RW	IrqDriver	0x0	Open-Drain
0x01	0-7	RO	Version	0x5	V. 5
		WO	WakeUpTimeByte	0x0	RTC Wake-Up Time Byte
0x02	0-7	RO	Revision	0x1	Revision Number of Digital Controller
	1-3	WO	WakeUpTimeWe	0x0	Wake-Up Time Byte Selector
0x03	1-4	RW	BattMgmtThreshold	0x0	No comparator is set
	7	RW	BattMgmtEnable	0x0	Disable Battery Management
	0	RO	BattMgmtCompare	0x0	Voltage is below the threshold value
0x04	0-3	RO	DioInValueAlarmStatus	0x0	No alarm
	0	WO	DioDirection	0x0	Digital IO port is configured as an input
	1	WO	DioOutValueAlarmEnable	0x0	Normal input
	2	WO	DioAlarmStart	0x0	No alarm
	3	WO	DioAlarmPolarity	0x0	Alarm on falling edge
	4	WO	DioUsePullup	0x0	No pull-up resistor
	5	WO	DioUsePulldown	0x0	No pull-down resistor
0x05	0-3	WO	DioPortWe		Writes settings of 0x04 in digital IO controller
0x06	0	RW	EnableWakeUpRtc	0x0	Disable Real Time Clock as wake-up source
	1	RW	EnableWakeUpDio	0x0	Disable Digital IOs as wake-up source
	4-6	RW	PowerUpTime	0x0	PowerUpTime128Ticks_C
	7	RW	PowerDownMode	0x0	Power Down Mode FULL
0x07	1	RW	ResetBbClockGate	0x1	Active reset of baseband clock distribution circuitry
	2	RW	ResetBbRadioCtrl	0x1	Active reset of the transceiver
	0	WO	PowerDown	0x0	Don’t shut down the chip
	6	WO	UsePullup4Test	0x0	No pull-up resistor - INTERNAL
	7	WO	UsePulldown4Test	0x0	No pull-down resistor - INTERNAL
0x08	0	RW	EnableBbCrystal	0x0	internal oscillator power off
	1	RW	EnableBbClock	0x0	baseband clock distribution off
	3	RW	BypassBbCrystal	0x0	Disable external oscillator
	4-6	RW	FeatureClockFreq	0x0	NA_FeatureClockDiv128_C

Table 179. List of default register settings (continued)

Register	Bit	RW	Field	Default	Description
	7	RW	EnableFeatureClock	0x0	Disable Feature Clock
0x09	0	WO	UsePullup4Spiclk	0x0	No pull-up resistor connected to the SPI clock pad
	1	WO	UsePulldown4Spiclk	0x0	No pull-down resistor connected to the SPI clock pad
	2	WO	UsePullup4Spissn	0x0	No pull-up resistor connected to the SPISsn pad
	3	WO	UsePulldown4Spissn	0x0	No pull-down resistor connected to the SPISsn pad
	4	WO	UsePullup4Spirxd	0x0	No pull-up resistor connected to the SPIRxD pad
	5	WO	UsePulldown4Spirxd	0x0	No pull-down resistor connected to the SPIRxD pad
	6	WO	UsePullup4Spitxd	0x0	No pull-up resistor connected to the SPITxD pad
	7	WO	UsePulldown4Spitxd	0x0	No pull-down resistor connected to the SPITxD pad
0x0A	0	WO	UsePullup4Por	0x0	No pull-up resistor connected to POR Pad
	1	WO	UsePulldown4Por	0x0	No pull-down resistor connected to POR Pad
	2	WO	UsePullup4Pamp	0x0	No pull-up resistor connected to TxRx Pad
	3	WO	UsePulldown4Pamp	0x0	No pull-down resistor connected TxRx Pad
	4	WO	UsePullup4Ucirq	0x0	No pull-up resistor connected to µcIRQ Pad
	5	WO	UsePulldown4Ucirq	0x0	No pull-down resistor connected to µcIRQ Pad
	6	WO	UsePullup4Ucrst	0x0	No pull-up resistor connected to µcRST Pad
	7	WO	UsePulldown4Ucrst	0x0	No pull-down resistor connected to µcRST Pad
0xB	0	WO	WritePulls4Spi	0x0	Write Pull-up and Pull-down settings to the four SPI Pads
	1	WO	WritePulls4Pads	0x0	Write Pull-up and Pull-down settings to the four additional Pads
0x0C	0-3	RW	TestModes	0x0	Default setting - Switches off transceiver test mode - INTERNAL
0x0D	0	WO	RfTestSelect	0x0	NA_RfTestSelTxDacCmd_C - INTERNAL
0x0E	0-1	RW	RamIndex	0x0	Baseband RAM block 0
	4-5	RW	DeviceSelect	0x0	Select baseband RAM
0x0F	0	RW	TxIrqEnable	0x0	TX interrupt does not drive the interrupt line
	1	RW	RxIrqEnable	0x0	RX interrupt does not drive the interrupt line
	2	RW	BbTimerIrqEnable	0x0	BbTimer interrupt does not drive the interrupt line
	3	RW	LoIrqEnable	0x0	LO interrupt does not drive the interrupt line
	4	RO	TxIrqStatus	0x0	Event did not trigger
	5	RO	RxIrqStatus	0x0	Event did not trigger
	6	RO	BbTimerIrqStatus	0x0	Event did not trigger

Table 179. List of default register settings (continued)

Register	Bit	RW	Field	Default	Description
	7	RO	LoIrqStatus	0x0	Event did not trigger
0x10	0-5	RO	TxIntsRawStat		Transmitter Interrupt Raw Status
	0-5	WO	TxIntsReset		Transmitter Interrupt Reset
	0-1		TxBufferRdy	0x0	Transmit Buffer Ready
	2		TxEnd	0x0	Transmit End
	3		TxUnderrun	0x0	Transmit Buffer Underrun
	4		TxTimeSlotTOut	0x0	Transmit Time Slot Time-out
	5		TxTimeSlotEnd	0x0	Transmit Time Slot End
0x11	0-6	RO	RxIntsRawStat	0x0	Receiver Interrupt Raw Status
	0-6	WO	RxIntsReset		Reset Receiver Interrupt
	0-1		RxBufferRdy	0x0	Receive Buffer Ready
	2		RxEnd	0x0	End of Packet Reception
	3		RxHeaderEnd	0x0	End of Correct Header Reception
	4		RxOverflow	0x0	Receive Buffer Overflow
	5		RxTimeSlotTOut	0x0	Receive Time Slot Time-out
	6		RxTimeSlotEnd	0x0	Receive Time Slot End
0x12	1	RO	LoIntsRawStat	0x0	Local Oscillator Interrupt Raw Status
	1	WO	LoIntsReset		Reset Local Oscillator interrupt
	7	WO	ClearBasebandTimerInt	0x0	Baseband timer interrupt reset
	1		LoTuningReady	0x0	Local Oscillator Tuning Ready
0x13	0-5	WO	TxIntsEn	0x0	Enable Transmitter Interrupt
	0-1		TxBufferRdy	0x0	Transmit Buffer Ready
	2		TxEnd	0x0	Transmit End
	3		TxUnderrun	0x0	Transmit Buffer Underrun
	4		TxTimeSlotTOut	0x0	Transmit Time Slot Time-out
	5		TxTimeSlotEnd	0x0	Transmit Time Slot End
0x14	0-6	WO	RxIntsEn	0x0	Enable Receiver Interrupt
	1-0		RxBufferRdy	0x0	Receive Buffer Ready
	2		RxEnd	0x0	End of Packet Reception
	3		RxHeaderEnd	0x0	End of Correct Header Reception
	4		RxOverflow	0x0	Receive Buffer Overflow
	5		RxTimeSlotTOut	0x0	Receive Time Slot Time-out
	6		RxTimeSlotEnd	0x0	Receive Time Slot End
0x15	1	WO	LoIntsEn		Enable Local Oscillator interrupt
	1		LoTuningReady	0x0	Local Oscillator Tuning Ready

Table 179. List of default register settings (continued)

Register	Bit	RW	Field	Default	Description
0x16-0x18	0-21	RW	LoRxCapsValue	0x200040	Read or Write the Caps Value for RX
0x19-0x1B	0-21	RW	LoTxCapsValue	0x200040	Read or Write the Caps Value for TX
0x1C	0	WO	LoEnableFastTuning	0x0	Disable Accelerated LO Caps Tuning
	1-3	WO	LoFastTuningLevel	0x0	Lowest LO Caps Tuning Level
	4	WO	LoEnableLsbNeg	0x0	No Higher Precision Accelerated LO Caps Tuning
	7	WO	UseLoRxCaps	0x0	Use TX caps for LO caps tuning
0x1D-0x1E	0-16	RW	LoTargetValue		target value for center frequency
0x1F	0-7	WO	AgcThresHold1	0x3	lower level of the integration value is 3
0x20	0-7	WO	AgcThresHold2	0x6	Upper level of the integration value: AgcThreshHold1 + 6
0x21	0-6	WO	HoldAgcInBitSync	0x18	“In bit synchronization state, gain is not changed after 24 pulses are received”
	7	WO	HoldAgcInFrameSync	0x1	Hold AGC when the receiver reaches frame synchronization state
0x22	0-5		AgcDeadTime	0xC	12 cycles before the next integration phase - INTERNAL
	6-7	WO	AgcNregLength	0x0	Gain Length N = 5
0x23-0x24	0-12	WO	AgcIntTime	0x200	512 cycles for AGC integration
0x25	0-5	WO	AgcValue	0x3F	AGC value used for RX amplifiers is 64
	6	WO	AgcDefaultEn	0x0	Ignore default Agc Value
	7	WO	AgcHold	0x0	RF AGC is active
0x26	0-5	RO	AgcGain	0x0	Read GAIN for Current Packet
	0-5	WO	AgcRssiThres	0x1E	“When gain is below 31, media is interpreted as busy when the upper bit of the TxPhCarrSenseMode field is set to 1”
	7	WO	AgcEnable	0x0	Disables AGC - INTERNAL
0x27	0-5	RO	FctPeriod	0x0	Counts the amount of 16MHz clock periods during a FCT clock cycle
	0-3	WO	ChirpFilterCaps	0x06	Switches the capacitors of the chirp filter to adjust the FCT clock to 400kHz
	4	WO	FctClockEn	0x0	RF TX FCT clock disabled
	5	WO	StartFctMeasure	0x0	RF TX FCT Measurement not active
	7	WO	EnableTx	0x0	Disable manual starting of TX
0x28-0x29	0-15	WO	BasebandTimerStartValue	0x0	Start value of the timer

Table 179. List of default register settings (continued)

Register	Bit	RW	Field	Default	Description
0x2A-0x2B	0-12	RO	ToaOffsetMeanAck	0x0	ToA Offset for Received Ack Packet - INTERNAL
	15	RO	ToaOffsetMeanAckValid	0x0	Reports that enough pulses were received at end of Ack packet - INTERNAL
0x2C-0X2D	0-15	RO	TxRespTime	0x0	TX Response Round Trip Time - INTERNAL
0x2E	0-2	RO	PhaseOffsetAck	0x0	PhaseOffset for Received Ack Packet - INTERNAL
	4-6	RO	PhaseOffsetData	0x0	PhaseOffset for Received Data Packet - INTERNAL
0x2F-0x30	0-12	RO	ToaOffsetMeanData	0x0	ToA Offset for Received Data Packet - INTERNAL
	15		ToaOffsetMeanDataValid	0x0	Reports if enough pulses were received to provide a valid time measurement - INTERNAL
0x2A-0x31	0-63	WO	SyncWord	0x0xAB69CA9492D52CAB	Inital Transceiver Syncword is 0x0xAB69CA9492D52CAB
0x31	0-3	RO	RxPacketType		Receive Packet Type
	4-5	RO	RxAddrMatch	0x0	Received destination address does not match the station address set in RamStaAddr0/1
	6	RO	RxCrc1Stat	0x0	CRC1 check has failed
	7	RO	RxCrc2Stat	0x0	CRC2 check has failed
0x32	0-3	RO	RxCorrBitErr	0x0	0 bit errors occurred during Syncword correlation
	4-7	WO	RxCorrErrThres	0x3	Maximum 3 syncword errors allowed
0x33-0x34	0-15	WO	TxTimeSlotStart	0x0	Defines the beginning of the time slot for a transmission
0x35-0x36	0-15	WO	TxTimeSlotEnd		Defines the end of the time slot for a transmission
0x33	0	RO	RxAddrSegEsMatch	0x0	End station address does not match
	1	RO	RxAddrSegIsMatch	0x0	Intermediate Station address does not match
	4	RO	RxCryptEn	0x0	Disable reading of the CryptEn Bit value
	5-6	RO	RxCryptId	0x0	Reads the value of the two bit Encryption ID field
	7	RO	RxCryptSeqN	0x0	Reads the value of the one bit sequential numbering scheme for encryption
0x34-0x35	0-14	RO	RxFec1BitErr	0x0	Reports the number of correctable single bit errors that have occurred in a receive packet
0x36		RO	Reserved		
0x37	0	WO	TxTimeSlotControl	0x0	Disable TX Time slot control - packets are transmitted independent of the time slot

Table 179. List of default register settings (continued)

Register	Bit	RW	Field	Default	Description
	1	WO	RxTimeSlotControl	0x0	Activates the receiver independent of the defined time slot through the RxCmd command
0x38-0x39	0-15	RO	RxPacketSlot	0x0	Reports time at the packet header end event
	0-15	WO	RxTimeSlotStart	0x0	Defines the beginning of the time slot for a reception
0x3A-0x3B	0-15	WO	RxTimeSlotEnd	0x0	Defines the end of the time slot for a reception
0x3C	0-3	RO	TxArqCnt	0x0	reports the required number of retransmissions
	4-7	WO	TxArqMax	0xE	Sets the maximal value for packet retransmissions
0x3D	0-1	WO	CsqDitherValue	0x0	no Chirp Sequencer dither
	2	WO	CsqUsePhaseShift	0x0	Disable Phase Shift
	3	WO	CsqUse4Phases	0x0	Disables four phases
	4	WO	ScqAsyMode	0x0	Disable Asymmetric Mode
	5	WO	CsqMemAddrInit	0x0	Disables initialize of chirp sequencer memory address
	6	WO	CsqUseRam	0x0	Disable Chirp Sequencer RAM
	7	WO	CsqTest	0x0	Disables Test RF Transmit Chirp - INTERNAL
0x3E	0-5	WO	CsqSetValue	0x0	Set Chirp Sequencer Value - INTERNAL
	6	WO	CsqSetIValue	0x0	Don’t set the I value in the Chirp Sequencer with the value programmed in CsqSetValue - INTERNAL
	7	WO	CsqSetQValue	0x0	Don’t set the Q value in the Chirp Sequencer with the value programmed in CsqSetValue - INTERNAL
0x3F	0	WO	D3IFixnMap	0x1	use FIX mode
	1	WO	D3IPomEn	0x0	Disables the mean value calculation in the Correlator
	2-3	WO	D3IPomLen	0x0	The last 2 values should be used for calculating the mean value
	7	WO	D3IUpDownEx	0x0	Disables exchange of Up and Down Pulses
0x40	0-6	WO	LeaveMapThresh1InBitsync	0x3	switch back to Fix mode after 3 detected pulses
	7	WO	UseMapThresh1InFramesync	0x0	use FIX with Threshold0
0x41	0	WO	Go2MapThresh1InBitsync	0x7	Switch from FIX mode using correlator threshold 0 to either MAP mode or FIX mode using correlator threshold 1 after 7 detected pulses in bit synchronization state.

Table 179. List of default register settings (continued)

Register	Bit	RW	Field	Default	Description
	1	WO	D3lFixThres1MapEn	0x0	Disable MAP mode or FIX mode using correlator threshold1 INTERNAL
0x42	0	WO	EnableLO	0x0	Disable RF local Oscillator
	1	WO	EnableLOdiv10	0x0	Disable LO frequency divide by 10 function
	2	WO	EnableCsqClock	0x0	Disable Chirp Sequencer Clock
	3	WO	InvertRxClock	0x0	Disable changing sampling Edge for receiver
	4	WO	EnableExtPA		Enable External Power Amplifier Output
	5	WO	EnableIntPA	0x0	Disable Power Internal Amplifier INTERNAL
	6	WO	EnableRxClock	0x0	Disable the receiver clock INTERNAL
	7	WO	EnableRX	0x0	Disable RF Receiver INTERNAL
0x43	0-2	WO	LnaFreqAdjust	0x3	Adjust RF Receiver LNA Frequency
	4-6	WO	TxPaBias	0x0	Adjust RF Transmitter Power Amplifier Bias
0x44	0-5	WO	TxOutputPower0	0x3F	Set RF TX Output Power0 to 63
0x45	0-5	WO	TxOutputPower1	0x3F	Set RF TX Output Power1 to 63
0x46	0-4	WO	RfRxCompValueI	0xF	Quantization Threshold for I
0x47	0	WO	RfRxCompValueQ	0xF	Quantization Threshold for Q
0x48	0-2	WO	SymbolDur	0x3	Set symbol duration to 4000ns
	4-6	WO	SymbolRate	0x7	Set Symbol Rate to 250k Symbols
	7	WO	ModulationSystem	0x0	Selects the 2-ary modulation system
0x49	0-1	WO	Crc2Type	0x0	Selects NA_Crc2type1_VC_C
	2	WO	UseFec	0x0	Disables FEC
	3	WO	TxRxCryptCrc2Mode	0x0	Enables NA_TxRxCryptCrc2ModeUncrypted_BC_C
	4-7	WO	TxRxCryptClkMode		Set Encryption/Decryption Clock Mode
0x4A	0	RW	SwapBbBuffers	0x0	Data in buffers belong to their intended direction
	1	WO	TxRxBbBufferMode1	0x0	Sets Duplex Mode
	2	WO	TxRxBbBufferMode0	0x0	Sets Auto Mode
	4	WO	FdmaEnable	0x1	Enables FDMA - 22 MHz bandwidth is used
	7	WO	TxRxMode	0x0	Sets Auto mode - MACFrame is built as a TN100 MACFrame with header information and data
0x4B-0x4C	0-2	WO	ChirpMatrix0	0x0	Downchirp
	4-6	WO	ChirpMatrix1	0x1	Upchirp
	8-10	WO	ChirpMatrix2	0x3	Plus Folded Chirp
	12-14	WO	ChirpMatrix3	0x4	Off Chirp

Table 179. List of default register settings (continued)

Register	Bit	RW	Field	Default	Description
0x4D	0-1	WO	TxPreTrailMatrix0	0x0	Downchirp
	2-3	WO	TxPreTrailMatrix1	0x1	UpChirp
	4	WO	TxUnderrunIgnore	0x1	Enable Underrun Ignore
	7	WO	TxMacCifsDis	0x0	Enable waiting for CIFS
0x4E	0	WO	TxVCarrSens	0x0	Virtual Carrier Sensing OFF
	1-2	WO	TxPhCarrSenseMode	0x0	Physical Carrier Sensing Mode off
	3	WO	TxVCarrSensAck	0x0	Disable Ack Packets for Virtual Carrier Sense
	4	WO	TxArq	0x1	Enable TX ARQ (Automatic Repeat Request) Error Correction Scheme for Data Packets
	5	WO	Tx3Way	0x0	Disable three-way handshake
	6	WO	TxBackOffAlg		Back-Off Algorithm
	7	WO	TxFragPrio	0x0	Disable Fragmentation Prioritization
0x4F	0-7	WO	TxBackOffSeed	0x0	Disable Back-Off Seed
0x50	0-3	WO	TxCryptSeqReset	0x0	Disable TX Decryption Sequence Reset
	4	WO	TxCryptEn	0x0	Disable Encryption
	5-6	WO	TxCryptId	0x0	Secret Key Selection for Encryption
	7	WO	TxCryptSeqN	0x0	Encryption Sequence Number
0x51	0-6	WO	TxScrambInit	0x7F	Initializes the value for the scrambling unit with 127
	7	WO	TxScrambEn	0x1	Enables scrambling in a transmitted MACFrame
0x52-0x53	0-12	WO	TxTransBytes	0x0	Number of Data Bytes in TX MACFrame in Transparent Mode
0x54	0-3	WO	TxPacketType	0x0	Packet Type For Transmission in Auto Mode
	7	WO	TxAddrSlct	0x0	Source Address for Transmission
0x55	0	WO	TxCmdStop	0x0	Don’t stop transmission
	1	WO	TxCmdStart	0x0	Don’t start transmission
	2-3	WO	TxBufferCmd	0x0	Transmit Buffer Ready for Transmission
0x56	0	WO	RxCmdStop	0x0	Don’t stop reception
	1	WO	RxCmdStart	0x0	Don’t start reception
	2	WO	RxBufferCmd	0x0	Read Receive Buffer
0x57	0-3	WO	RxCryptSeqReset	0x0	RX Decryption Sequence Reset
0x58-0x59	0-12	WO	RxTransBytes	0x0	Number of Data Bytes in Received MACFrame in Transparent Mode (0=8192bytes)
0x5A	0	WO	RxTimeBCrc1Mode	0x1	Enable Receive TimeB packets CRC1 check.

Table 179. List of default register settings (continued)

Register	Bit	RW	Field	Default	Description
	1	WO	RxCrc2Mode	0x1	RxEnd is only triggered when CRC2 check succeeds. When the check fails the receiver remains active to receive the next packet. When overflow occurs this will be ignored.
	2-3	WO	RxArqMode	0x2	The ARQ scheme is based on the CRC1 and CRC2. An Ack packet is sent in response to a Data packet only if CRC1, CRC2, and address matching succeeded.
	4	WO	RxAddrSegEsMode	0x0	Disables End Station Address Matching
	5	WO	RxAddrSegIsMode	0x0	Disable Intermediate Station Address Matching
	6-7	WO	RxAddrSegDevIdL	0x0	Segmented Address Format
0x5B	0	WO	RxDataEn	0x1	Enable Data packets Reception
	1	WO	RxBrdcastEn	0x1	Enable Broadcast Packet Reception
	2	WO	RxTimeBEn	0xq	Enable TimeB Packet Reception
	3	WO	RxAddrMode	0x1	Sets Address Matching Mode
	4-7	WO	RangingPulses	0x5	Use 24 Up and Down Pulses for ranging INTERNAL
0x5C	0-4	WO	PulseDetDelay	0x5	Delay for Pulse Detection is 5
0x5D	0-2	WO	GateAdjThreshold	0x7	sets the required number of consecutive pulses outside of the gate center to 7 before a readjustment of the gate takes place
	4	WO	DownPulseDetectDis	0x0	Enable Detection of Down Pulses
	5	WO	UpPulseDetectDis	0x0	Enable Detection of Up Pulses
0x5E	0-1	WO	GateSizeUnsync	0x1	Sets gate size = 5 slots for unsynchronized state
	2-3	WO	GateSizeBitsync	0x1	Sets gate size = 5 slots for bit synchronized state
	4-5	WO	GateSizeFramesync	0x1	Sets gate size = 5 slots for synchronized state
	6	WO	GateAdjBitsyncEn	0x1	Enables gate adjustment in bit synchronization state
	7	WO	GateAdjFramesyncEn	0x1	Enables gate adjustment in frame synchronization state
0x5F	0-2	WO	Go2BitsyncThreshold	0x2	Sets the required number of consecutive chirp pulses detected in the unsynchronized state to go to bit synchronization state to 2
	4-6	WO	LeaveBitsyncThreshold	0x6	Sets the required number of consecutive lost pulses in bit synchronization state to return to the unsynchronized state to 6
0x60	0-7	WO	RtcTimeBTxAdj	0x0	Before transmitting, the Real Time Clock control value must be adjusted to compensate for transmitter delays
0x61	0-7	WO	RtcTimeBRxAdj	0x0	Adjusts the Real Time Clock value after reception to compensate for receiver delays

Table 179. List of default register settings (continued)

Register	Bit	RW	Field	Default	Description
0x62	0	WO	RtcCmdWr	0x0	Do not Set RamRtcReg Value in Real Time Clock
	1	WO	RtcCmdRd	0x0	Don't Set Real Time Clock Value in RtcReg
	4	WO	RtcTimeBAutoMode	0x0	The value of RamRtcTx is transmitted in the RTC field of the TimeB packet, while the RTC value in a received TimeB packet is stored in RamRtcRx
	7	WO	RtcTimeBTestMode	0x0	Disables Real Time Clock TimeB Test Mode - INTERNAL
0x63	0-4	WO	AgcAmplitude	0x0	Amplitude of 0 is selected for alternative AGC
0x64	0-3	WO	AgcRangeOffset	0x0	AGC is temporarily halted when: | Mean Value - Middle | > 0
	7	WO	UseAlternativeAgc	0x0	Disables Alternative AGC
0x65-0x7C			Reserved	0x0	
0x7D	0	RW	TxRxDigTestMode	0x0	Transceiver Digital Test Mode - INTERNAL
	1	RO	DebugMacRxCmd	0x0	Debug Mac Command is idle - INTERNAL
	2	RO	DebugMacTxCmd	0x0	Debug Mac Command is idle - INTERNAL
0x7E	0-3	RO	DebugMacFsm	0x0	Debug Mac Fsm is idle - INTERNAL
	4-7	RO	DebugMacTxRxCtrl	0x0	Debug MAC Transmit Receive Control is idle - INTERNAL
0x7F	0	RO	DebugBitProcFsm	0x0	Debug Bit Process Fsm is idle - INTERNAL
	1	RO	DebugBitProcTx	0x1	Debug Bit Processing Unit is idle - INTERNAL
	2	RO	DebugBitProcRx	0x0	Debug Bit Processing Unit is idle - INTERNAL
	3	RO	DebugRxDetStatus	0x1	Debug Receiver Detector Status unsynchronized - INTERNAL

Revision history

Table 180. Document revision history

Date	Revision	Changes
05-Sep-2008	1	Initial release.