ZL70550 Programmer User's Guide
for
ZL70550 Ultra-Low-Power Sub-GHz RF Transceiver

Microsemi.

Power Matters.™

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
E-mail: sales.support@microsemi.com
www.microsemi.com

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided “as is, where is” and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such informvdation. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.

About Microsemi

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif., and has approximately 4,800 employees globally. Learn more at www.microsemi.com.

Contents

Figures

Tables

1 Revision History

The revision history describes the changes that were implemented in the ZL70550 Programmer User's Guide (153552). The changes are listed by revision, starting with the most current publication.

1.1 Initial Release History

Revision 1, dated July 2016, was the first publication of this document.

2 Overview

2.1 Introduction

The ZL70550 Programmer User’s Guide contains a comprehensive list of typical and required programming procedures for the various modes of operation and required calibrations of the ZL70550 Ultra-Low-Power Sub-GHz RF Transceiver. Complementing these procedures is a complete memory map defining all of the application registers, with detailed descriptions of their bit definitions including reset values and register access types.

For programming examples, example source code written in C is available to all users who complete a Source Code License Agreement (SCLA) with Microsemi. This source code provides examples of many of the procedures in this user’s guide and therefore significantly reduces the development time for users. This source code runs on the ZL70550 Application Development Kit (ADK) boards, which is also available and recommended for users. The ZL70550 ADK provides users with a platform to observe the behavior of the procedures in a lab environment. The ZL70550 ADK also provides an example circuit, allowing users to evaluate the RF performance of the device.

2.2 Product Description

Microsemi’s ZL70550 RF transceiver is a low-power sub-GHz ISM-band radio designed for wireless-sensor applications that use either continuous monitoring or low-duty-cycle monitoring. The ZL70550 device operates in unlicensed frequency bands between 779 MHz and 965 MHz and offers a maximum data rate of 200 kbit/s to support voice communication. For data communication, the ZL70550 supports extremely low power consumption in packet-based networks. The device includes the RF transceiver as well as a Media Access Controller (MAC) that performs most link support functions.

The ZL70550 system uses the same device at both ends of the communication link.

All control and packet transfers occur across the SPI interface. Setup and control information is accessed with SPI interface operations to either volatile or always-on memory-mapped registers. Packet data is also transferred with block read/write operations.

After the device is set up and necessary information is written to the registers, packet operations can be performed. Packets are written to the transmit buffer across the SPI interface, with or without packet headers. If written without headers, then the headers are automatically added to the packets, depending on the packet format. Likewise, packets can be read from the receive buffer, with or without packet headers. The packet headers are decoded by the device as necessary per the packet format.

Operations can be performed as single-packet operations or, alternatively, the device can perform intelligent packet sequence transactions. With packet transactions, automatic packet acknowledgments are transmitted and received, and if the acknowledgment is not received, automatic retransmissions are attempted.

Completion of packet operations is signaled by various interrupts, depending on the operation being performed, at which point the host processor is notified of the result of the operation.

The device is intended to operate primarily in CSMA mode, but can also support TDMA operation. In CSMA mode, the node initiates all packet transfers starting with a CSMA-CA operation to detect whether the channel is clear. It then transmits that packet and typically waits for an acknowledgment packet. If the acknowledgment fails, then the node may attempt a retransmission after a random back-off period and another CSMA-CA operation to detect whether the channel is clear.

The hub typically waits in receive mode for a packet. Once a packet is received, it transmits a response. In Z-Star packet mode, the hub may also immediately send another nonacknowledgment packet following the acknowledgment packet by signaling in the acknowledgment packet that it will be doing so.

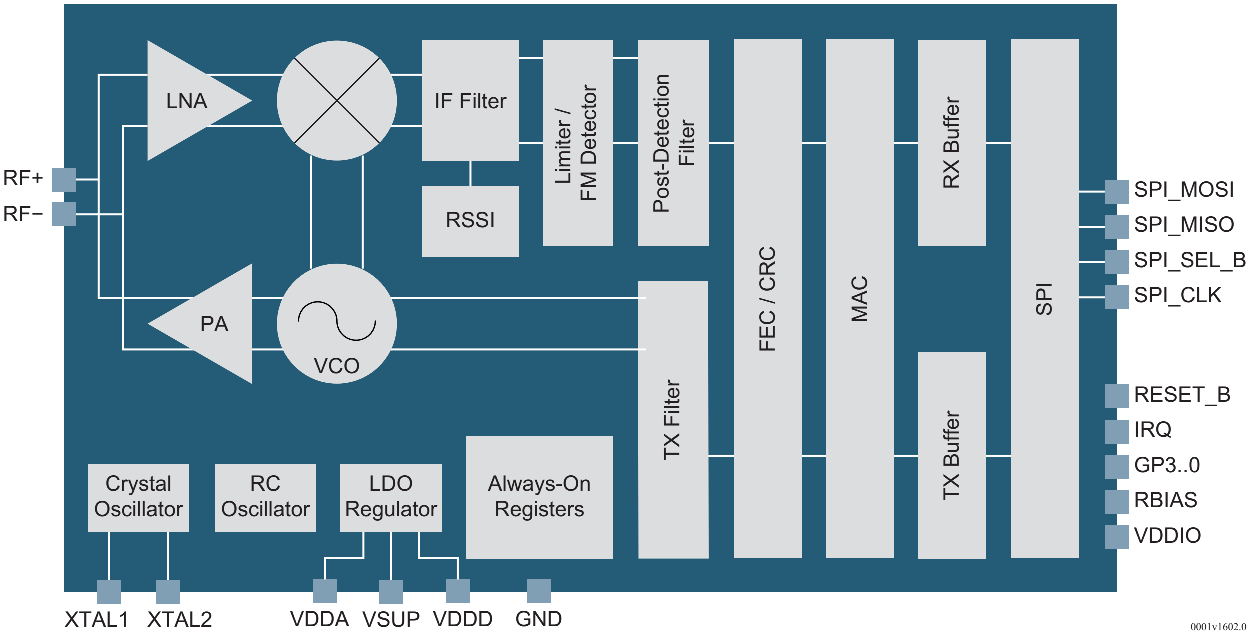
Typically in CSMA operation, the hub does not initiate packet transfers and must wait for the node to send a packet. For nodes that do not periodically transmit data, they must periodically send data-request packets to see if the hub has a packet for them.

3 Architecture

3.1 Block Diagram

The ZL70550 block diagram is shown in Figure 1, page 3. All control and data is transferred across the SPI slave interface. The 2,048-byte internal address space includes status and control registers plus a 1,024-byte packet buffer.

Figure 1 • **ZL70550 Block Diagram**



3.2 System Clock

The system clock is derived from the crystal oscillator (XTAL) clock, running at 24 MHz and giving the frequencies shown in Table 1, page 3.

Table 1 • **Default Frequencies**

Crystal	24.0 MHz
Internal system clock	1.20 MHz
PLL clock channel spacing	300 kHz
IF frequency	600 kHz
Supported bit rates	200 kbit/s,100kbit/s, and 50kbit/s

4 Communication Protocol

4.1 Packet Definition

The packet format provides a high effective data rate with excellent error detection and error correction capability. As shown in Table 2, page 4, depending on the packet mode selected by the user, the format of a complete packet may contain:

- Packet preamble
- Frame sync
- PHY header (Z-Star packet mode and user packet mode only)
- MAC header (Z-Star packet mode only)
- User data blocks
- Frame Check Sequence (FCS) or CRC

This packet format is used in both directions, from hub to node and from node to hub. For maximum data throughput, the packet preamble and header are both as small as possible, while allowing up to 511 bytes of user data within a packet.

Sections 4.1.1.1 Z-Star Packet Mode, page 5, through 4.1.1.3 Raw Packet Mode Format, page 6, show the structure of the complete packet: the preamble, the header, a typical data block, and CRC. The time sequence runs from left to right, so the preamble is first and CRC is last.

4.1.1 Packet Modes Overview

All packet transfers occur to and from the internal packet buffer across the system bus between the SPI interface and the MAC. Typically, 512 bytes are allocated to the transmit buffer, and 512 bytes to the receive buffer. Packet sizes are typically limited to 511 bytes.

The packet mode is selected using *tx_mode* in the **TX_CTRL0** register for TX and using *rx_mode* in the RX_CTRL0 register for RX. The various packet modes provide different features, as shown in Table 2, page 4.

Table 2 • **Packet Modes of Operation**

Mode	Description	Frame Sync	FEC	PHY Header	Auto-Length	MAC Header	CRC
Raw bit mode	Optional serial clock and data (TX/RX buffer or **GP3..0** pins)	No	No	No	No	No	No
Raw packet mode	Compatible with ZL70251 MAC with optional FEC and CRC	Yes	Opt	No	No	No	Opt
User packet mode	User-defined packet (no MAC header)	Yes	Opt	Yes	Yes	No	Opt
Z-Star packet mode	Fully functional MAC based on Microsemi’s Z-Star protocol	Yes	Opt	Yes	Yes	Yes	Yes

There are restrictions on CRC availability in raw packet mode, since there is no length information in the packet. Either fixed length packets must be used, or length information must be extracted dynamically during packet reception to set the length.

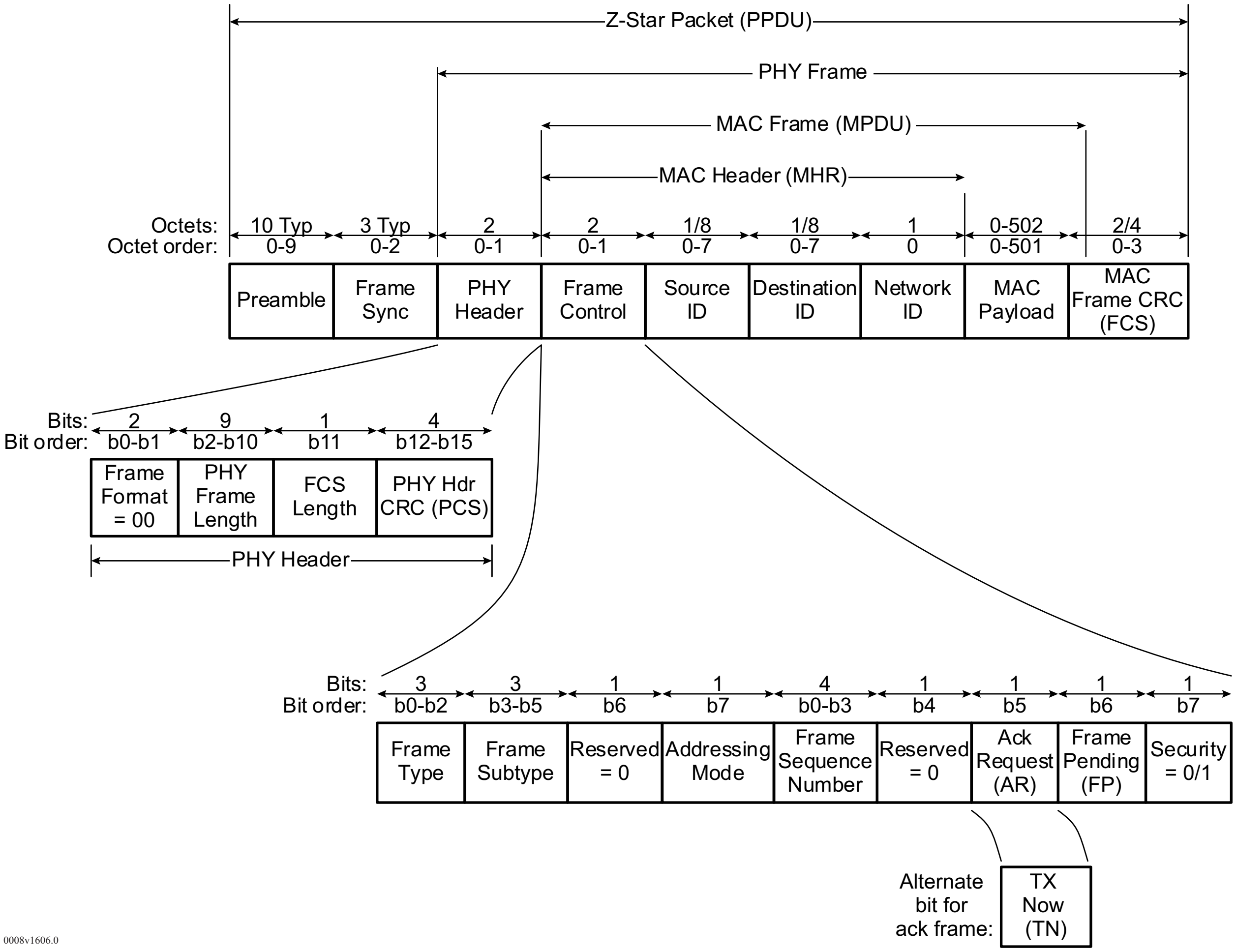
4.1.1.1 Z-Star Packet Mode

The function of the transmitter is to build the packet PHY frame, as shown in Figure 2, page 5. The recommended preamble length is 10 bytes, and the frame synchronization pattern is programmable from 2 to 5 bytes. The Z-Star packet is composed of six distinct sections for the Z-Star frame:

- Preamble
- Frame synchronization pattern
- PHY header
- MAC header
- Payload
- Frame Check Sequence (FCS) or CRC

The generated PHY frame is optionally hamming encoded and serialized in the MAC for final packet generation and transmission.

Figure 2 • **Packet Format, Z-Star Packet Mode**



4.1.1.2 User Packet Mode

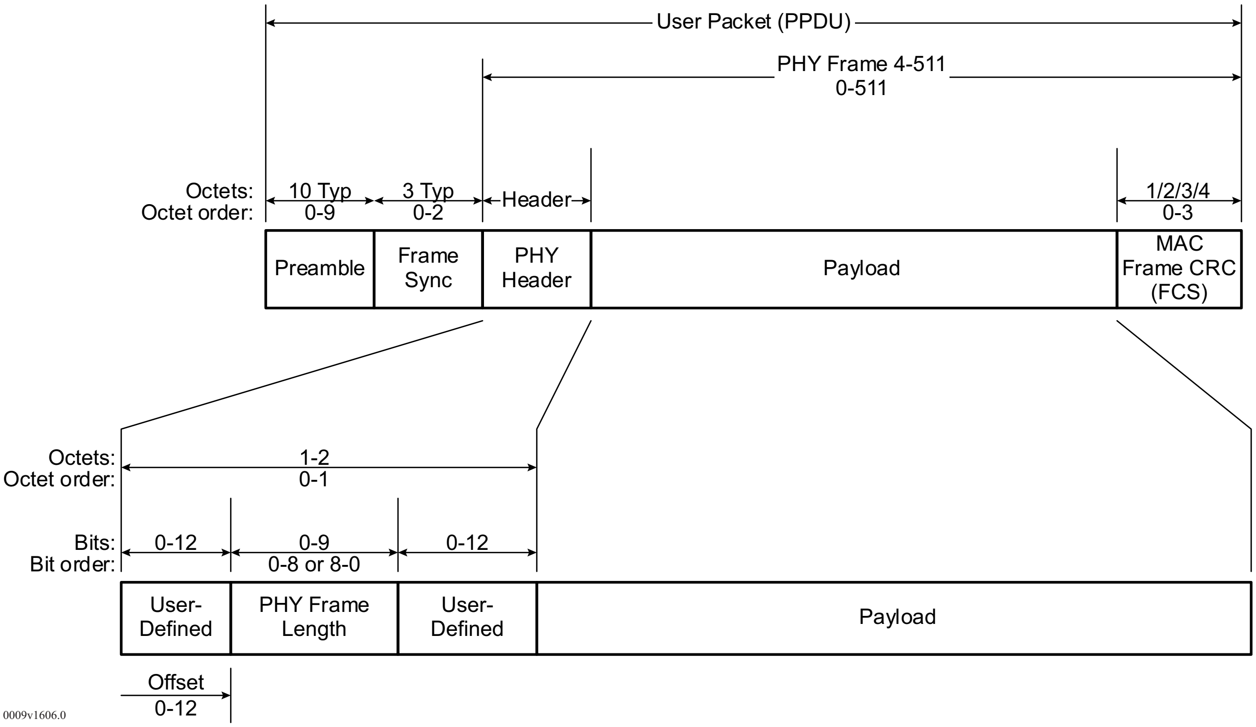
In the user packet mode, the frame is composed of the PHY header, payload, and FCS as shown in Figure 3, page 6. There is no MAC header. The FCS is variable, set by *rx_fcs_len* for receive and *tx_fcs_len* for transmit. If the setting is zero, then there is no FCS. The correct FCS polynomial must be set in *crc_poly*. Both values must be the same for both devices in any one direction, but can be different for the different directions.

The recommended preamble length is 10 bytes, and the frame synchronization pattern is programmable from 2 to 5 bytes.

For transmit, the length is generated automatically by the SPI packet write command if *tx_auto_hdr* is set to 1; in user packet mode, this limits the PHY header to one byte and therefore limits frame length to 255 bytes. Alternatively, if *tx_auto_hdr* is set to 0, a two-byte PHY header is an option and the length is determined by *tx_buf_len*, so the maximum packet size is 511 bytes. In the latter case, the user must write the entire packet, including the frame length in the PHY header, to the transmit buffer.

The frame length may be placed anywhere in the first two bytes, but the receiver must preprogrammed for the location of the frame length field.

Figure 3 • **Packet Format, User Packet Mode**



4.1.1.3 Raw Packet Mode Format

The raw packet mode is similar to the user packet mode, except that there is no PHY header and no frame length in the packet. In raw packet mode, packets are transmitted without a header.

The packet format is shown in Figure 4, page 7. The preamble and frame sync are transmitted, so that received data is byte-aligned before being written to the RX buffer. The operation in raw packet mode is limited to that shown in the state machine operation shown in Figure 15, page 24. Raw packet mode can be used if the packet format does not support the PHY header requirements that exist in user packet mode for the packet length field. In this case, the TX and RX packet length are controlled by *tx_buf_len* and *rx_frm_len* respectively.

The recommended preamble length is 10 bytes, and the frame synchronization pattern is programmable from 2 to 5 bytes. FEC encoding/decoding is optional. Preamble and frame synchronization length, and FEC selection operate the same as for other modes.

The RX frame length *rx_frm_len* may be updated during packet reception, providing this occurs before the end of the packet. The packet may optionally be terminated by a SPI abort command. In all cases, ***rxc_pkt_buf_len[8:0]*** indicates the number of bytes written to the RX buffer (which is either the entire PHY frame length or the payload length; see Table 10, page 27).

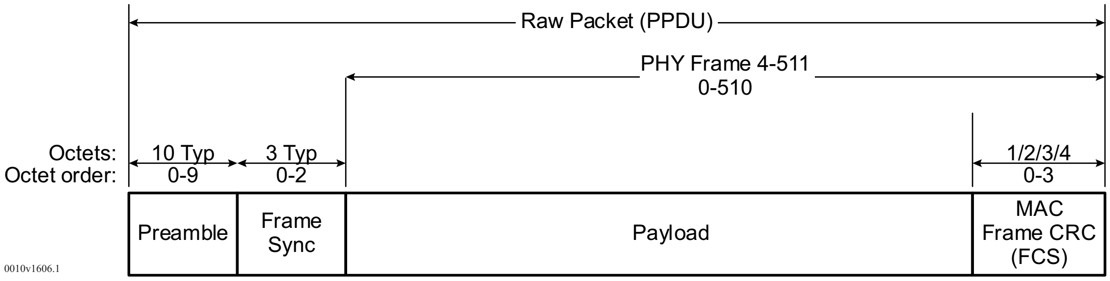
The FCS is variable, set by *tx_fcs_len* and *rx_fcs_len*. If the setting is zero, then there is no FCS. The correct FCS polynomial must be set in *crc_poly*. Both values must be the same for both devices in any one direction, but can be different for the different directions.

The CRC is optional and requires using one of two methods for RX packet length.

- A fixed packet length is used, in which case the CRC is always known.
- The host CPU dynamically derives the length from the contents of the beginning of the packet, and writes that value to *rx_frm_len* prior to the end of packet being received. Otherwise, the FCS cannot be calculated correctly.

To facilitate the latter case, a programmable interrupt, *rx_hdr_rdy_irq*, is available to interrupt the host CPU after a predetermined number of packet bytes have been written to the RX buffer. The RX buffer fullness threshold *rx_hdr_thresh* controls the number of bytes written to the RX buffer, at which point *rx_hdr_rdy_irq* is asserted.

Figure 4 • **Packet Format, Raw Packet Mode**



4.1.2 Frame Control

4.1.2.1 Frame Control for Raw Bit Mode Operation

In raw bit mode, raw bits are transmitted from the transmitter without preamble, frame sync pattern, header, or CRC. If these properties are needed, then they must be encoded in the bit stream.

In raw bit TX mode, the transmitter sends a user defined bit stream. The source of the data can be either the TX buffer or the **GP1** pin. It is recommended that the preamble be included in the data stream, unless manual DC receiver settings can be used.

On the receiver side, the bit stream is input without frame synchronization or byte alignment. The received data is placed in the receive buffer, and can also be output to a general-purpose I/O pin along with an output clock.

In raw bit mode, the receiver does not require a frame sync to receive the RX data. As soon as the receiver is enabled, data is streamed into the RX buffer. All data is streamed to the receive buffer, with no byte alignment and no dewhitening. Optionally, the data could be qualified by preamble detection.

Raw bit mode has two basic applications. First, is can be used for raw bit error testing using the GPIO TX and RX I/O modes. Second, it can be used for applications where the packet framing is not desired, or for data rates not supported by the ZL70550. There are limitations to this second case.

This mode allows for user defined bit rates, particularly if **GP1** is used for input data, because the user can change the bit value at his chosen rate. On the receiver side, the raw RX data bits are available either in the RX buffer or at a general-purpose I/O pin.

To use raw bit mode, refer to the setup and initialization information provided in Section 6.5.2 For Raw Bit Mode, page 38.

4.1.2.2 Frame Control for Packet Mode Operation

Packet mode is used in Z-Star packet mode, user packet mode, and raw packet mode.

In packet mode, preamble and frame sync is added during transmissions, and expects both preamble and frame sync during receive. For optimum performance of the receiver, the DC level of the receiver must be set, and the preamble is used to automatically set the DC level. The frame sync is needed to perform byte alignment and synchronize the whitener in the PHY. It is also used in the MAC to locate the beginning of the PHY frame, which is required for FEC decoding, and header decoding, and CRC checking.

4.1.3 Frame Processing

4.1.3.1 Frame Processing in Z-Star Packet Mode

In Z-Star packet mode, frame processing is performed per Figure 14, page 23. This state machine diagram shows the sequence of packet transfers that can be performed automatically by the device, and follows the transaction sequences defined by the Z-Star protocol.

Figure 5 • **Basic Transaction**

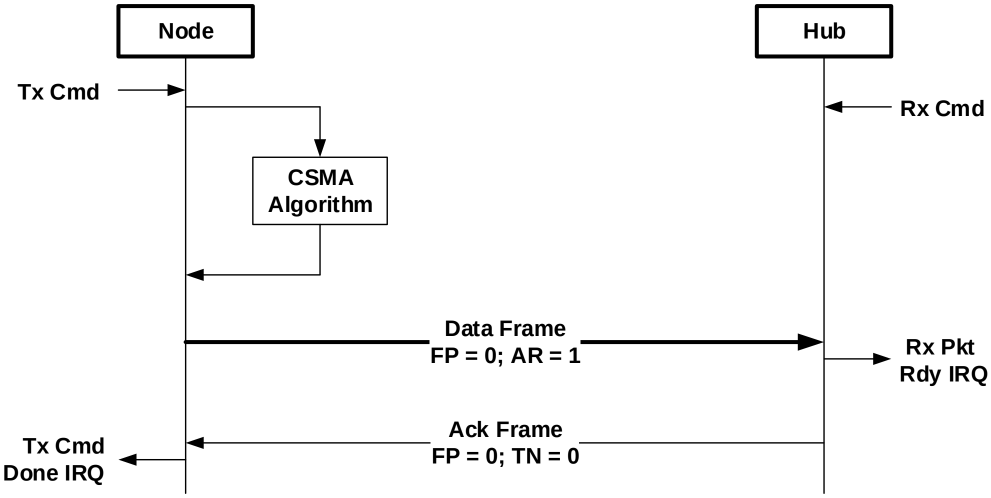


Figure 5, page 8, shows the basic packet transaction. Typically the node starts in CSMA1 to listen for a clear channel before proceeding to TX1 to transmit a packet or send a data request. After sending a non-data-request packet, it turns around to RX to receive the acknowledgment packet. The node is interrupted with a *cmd_done_irq* when the packet operation is complete.

The hub remains in blind reception receive mode, waiting for a packet from any node in its network. When a packet is received, as indicated by the *rx_pkt_rdy_irq* and the AR bit is set, then the hub sends an acknowledgment packet back to the node. The FP and TN bits are set to 0 in the acknowledgment packet to indicate to the node that the hub has no packets for the node. This completes the packet transaction.

If the packet is not received by the hub, or if the acknowledgment packet is sent and not received by the node, then the node retransmits the same packet, as shown in Figure 6, page 9. This start with a random back-off prior at the beginning of the CSMA operation, followed by the retransmission of the packet.

If the hub needs to transmit a packet to the node, it must wait for that node to initiate either a packet transmission, or a data-request packet transmission. The data packet transmission is shown in Figure 7, page 9. When the hub receives a packet, it must check its transmit queue for that node to see if it has a pending packet for that node. If it does, the hub sets FP and TN to 1 in the acknowledge packet, signifying to the node that it will immediately receive another packet.

When this packet is received, the node is notified with the *rx_pkt_rdy_irq*, and if the AR bit is set in the packet header, the node sends the acknowledgment packet back to the hub.

When the hub receives the acknowledgment as signaled by *rx_ack_done_irq*, it can mark that packet as sent and remove it from its transmit queue for that node.

Figure 6 • **Basic Transaction with Retry**

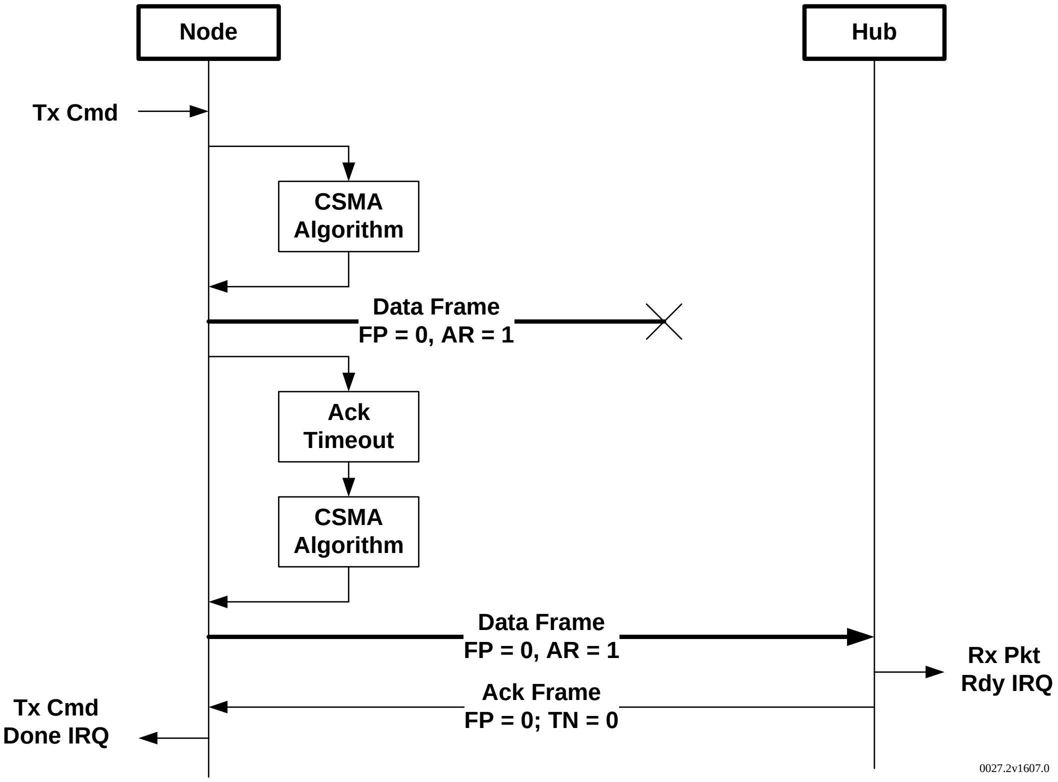
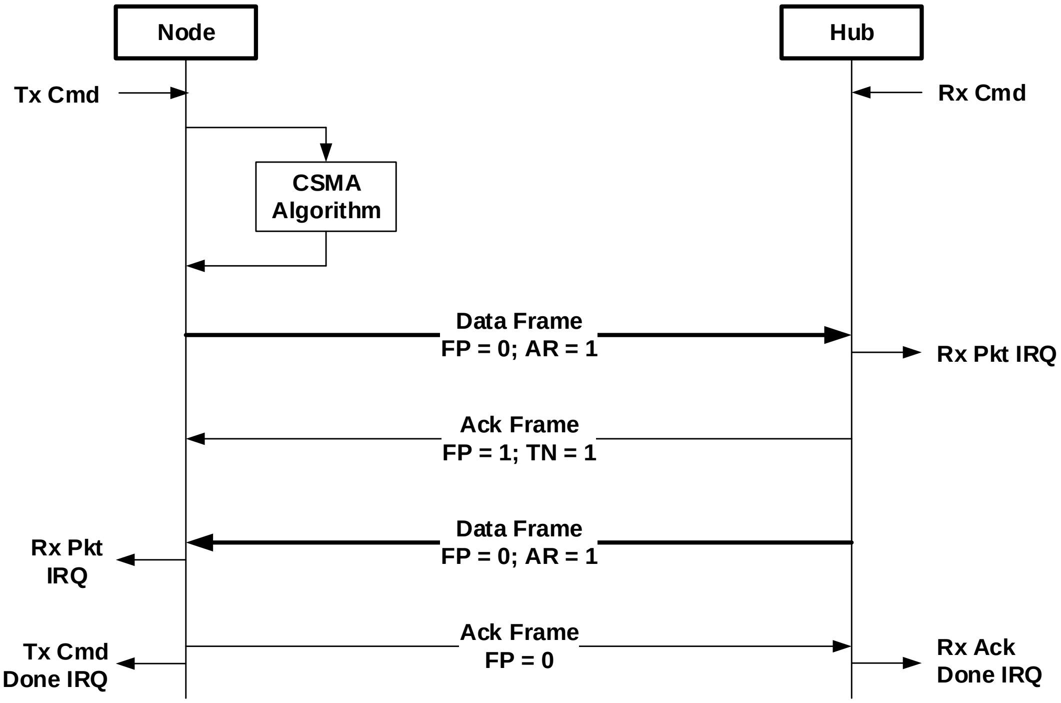


Figure 7 • **Hub to Node Frame Transaction**



The node should periodically communicate with the hub, so that the hub and node remain connected and so the hub can send packets to the node. If the node does not have periodic data for the hub, then it should periodically send a data-request packet to the hub, as shown in Figure 8, page 10.

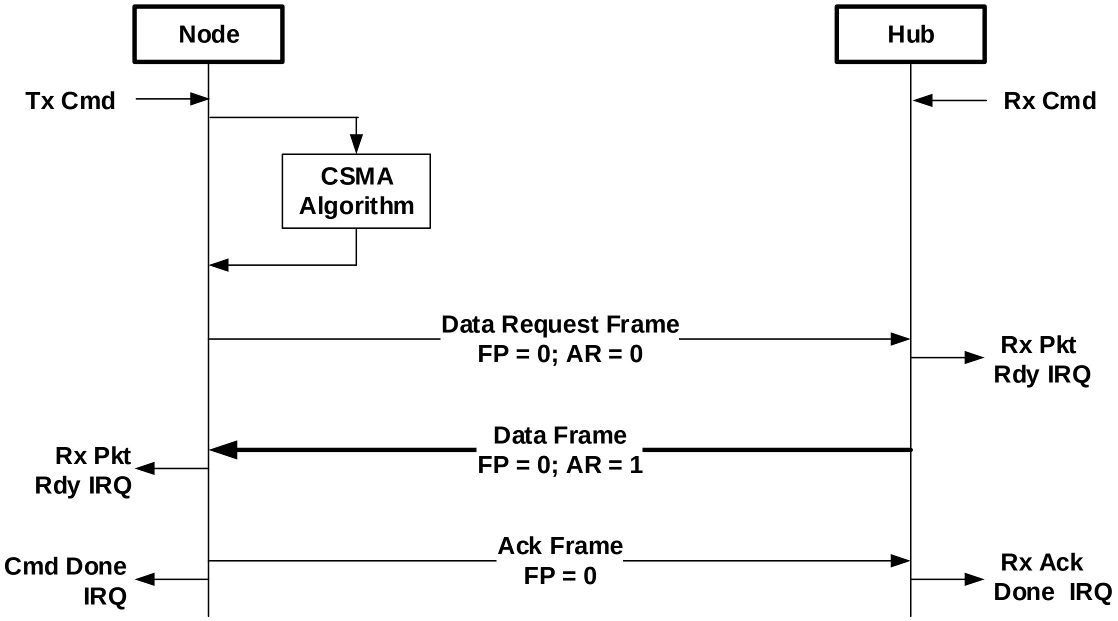
The node first checks for a clear channel with CSMA, and then sends a data-request packet. The data-request packet contains the next expected frame sequence number for that node.

When the hub receives the data request, it must check its transmit queue to see if it has a packet for that node. If so, it immediately sends the packet with the expected frame sequence number, typically with the AR bit set to request an acknowledgment packet.

When the node receives the packet, it is interrupted with *rx_pkt_rdy_irq*, and the acknowledgment packet is automatically transmitted back to the hub. When the hub receives the acknowledgment, it clears the packet from its transmit queue for that node.

If the hub does not have a packet for that node, or if the hub has already transmitted the requested sequence number, then it sends a null packet with AR equal to 0. This terminates the transaction on both sides.

Figure 8 • **Data Request Transaction**



4.1.3.1.1 Z-Star FCS Calculation

In Z-Star packet mode, the polynomial is controlled dynamically, depending of the FCS length. The transmitter transmits an FEC length as defined in *tx_fcs_len*. If set to 3’b010, then the 16-bit polynomial is used for calculating the FCS. On the receiver, if FCS length is set to 0 in the PHY header, the 16-bit FCS is calculated over the entire MAC frame, not including the PHY header. The 16-bit FCS polynomial is shown in EQ 4-1. The polynomial is entered LSB first, also known as reverse bit order. For 16-bit FCS, the polynomial is 0x00008408.

$$FCS = x^{16} + x^{12} + x^5 + 1 \qquad \textit{EQ 4-1}$$

If *tx_fcs_len* is set to 3’b100, then the transmitter uses to 32-bit polynomial to generate the 32-bit FCS. On the receiver, it FCS length in the PHY header is set to 1, the 32-bit FCS is calculated over the entire MAC frame, not including the PHY header, and uses the 32-bit FCS polynomial shown in EQ 4-2. For the 32-bit FCS, the polynomial is 0xEDB88320.

$$FCS = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1 \qquad \textit{EQ 4-2}$$

4.1.3.1.2 Z-Star Header Generation

The MAC header may be automatically generated from the always-on registers, or loaded into the transmit buffer, depending on the setting of *tx_auto_hdr*, It is set to 1 for auto-header generation. In auto-header generation, the MAC header is generated from the contents of the always-on registers, from the previous received packet, or from default values per packet type. For acknowledgment, data request, and null packets, *tx_auto_hdr* is ignored and the headers are automatically generated, where the frame type and frame subtype fields, and other fields, are over-ridden relative to the values in the always-on registers.

Table 3, page 11, shows the typical header field sources for auto-header generation in transmit (where the column headings correspond to the state machine diagram shown in Figure 14, page 23). RX packet means that fields are derived from the current/last RX packet received. Payloads with payload lengths are only used in TX1 and TX2, where the command is not TX data request and when *tx_null_frm* is 0.

The contents of the packet may always be completely specified, over-riding the contents of Table 3, page 11, by two methods. One method is to use a SPI TX command (not TX data request) with *tx_auto_hdr* equal to 1 and *tx_null_frm* equal to 0, and have all settings in the always-on registers. The other method is to use a SPI transmit command with *tx_auto_hdr* equal to 0 and *tx_null_frm* equal to 0, and place the header in the transmit buffer.

Table 3 • **Auto Header Generation and Checking**

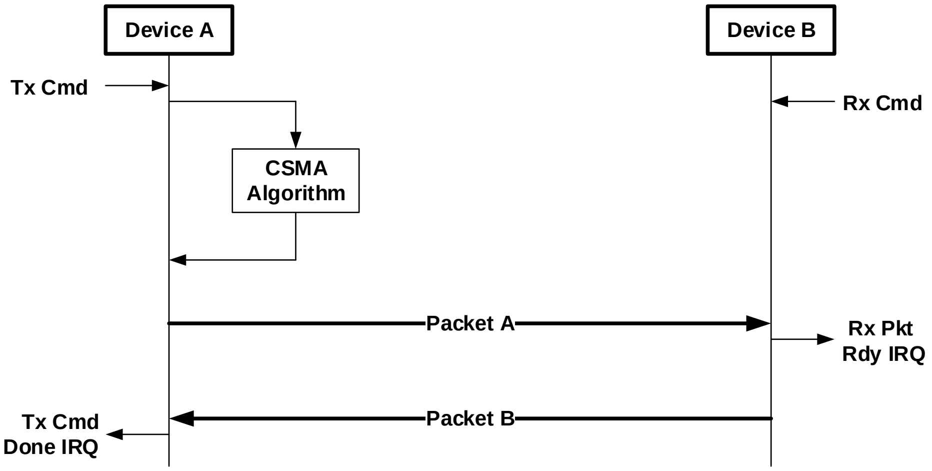
	TX1 Data Pkt	TX1 Null Pkt	TX1 Data Request	TX_ACK2	TX_ACK1	TX2
Source address	Always-on registers	Always-on registers	Always-on registers	Always-on registers	Always-on registers	Always-on registers
Destination address	Always-on registers	Always-on registers	Always-on registers	RX packet	RX packet	RX packet
Network address	Always-on registers	Always-on registers	Always-on registers	RX packet	RX packet	RX packet
Address mode	Always-on registers	Always-on registers	Always-on registers	RX packet	RX packet	RX packet
FSN	Always-on registers	0	1 + last RX FSN	RX packet	RX packet	Always-on registers
FP	Always-on registers	0	0	0	Always-on registers	Always-on registers
AR/TN	Always-on registers	0	0	0	Always-on registers	Always-on registers
FCS length	Always-on registers	2	2	2	2	Always-on registers
Frame type	Always-on registers	Data	Data request	Ack	Ack	Always-on registers
Payload length	Always-on registers	0	0	0	0	Always-on registers

4.1.3.2 Frame Processing in User Packet Mode

The user packet mode, processing is shown in Figure 15, page 24. As in Z-Star, the network is configured for CSMA operation with the node, Device A, initiating all packet transactions. When Device B receives a packet and is interrupted with *rx_pkt_rdy_irq*, it immediately turns around and starts transmitting the response packet. The host CPU must have the packet loaded into the transmit buffer by the time the frame synchronization pattern is transmitted. After the packet is transmitted from Device B, Device B goes back to receive mode, waiting for another packet.

When Device A receives a good packet, the transaction sequence is complete, and *cmd_done_irq* is asserted. This transaction sequence is shown in Figure 9, page 12. If the packet reception fails, then Device A performs a random back-off at the beginning of the CSMA and automatically retransmits the same packet up to the retry count. If no response packet is received, and all retries are exhausted, then the process exits with a *cmd_done_irq* interrupt and the *cmd_fail_irq* status.

Figure 9 • **Packet Processing in User Packet Mode and Raw Packet Mode**



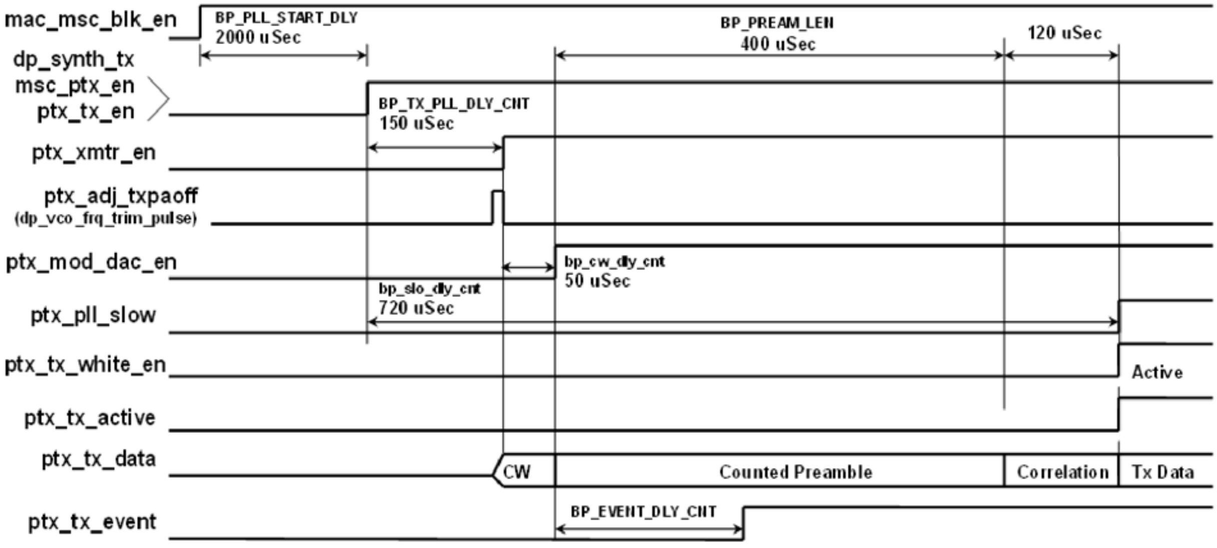
4.1.4 Packet Exchange Timing

There are several registers that control the timing in the ZL70550. This timing relates to the sequence of internal operations required to transmit or receive packets. Microsemi recommends using either the default values or the values listed in Section 8.2 Address Space, page 53, which were tested and chosen for optimal system performance.

4.1.4.1 TX Timing Delays (Default Register Settings)

Figure 10, page 12, shows the default TX start-up timing, using the default register settings. The *pll_start_dly* controls the setting time of the PLL start-up from when it is enabled. This typically occurs when *msc_blk_en* is high after wake-up. The CSMA, transmit, and receive operations cannot start until this delay is complete. It only occurs once during the start-up of the device after power-on. For other TX/RX start-up conditions, it is bypassed.

Figure 10 • **Default TX Start-Up Timing**



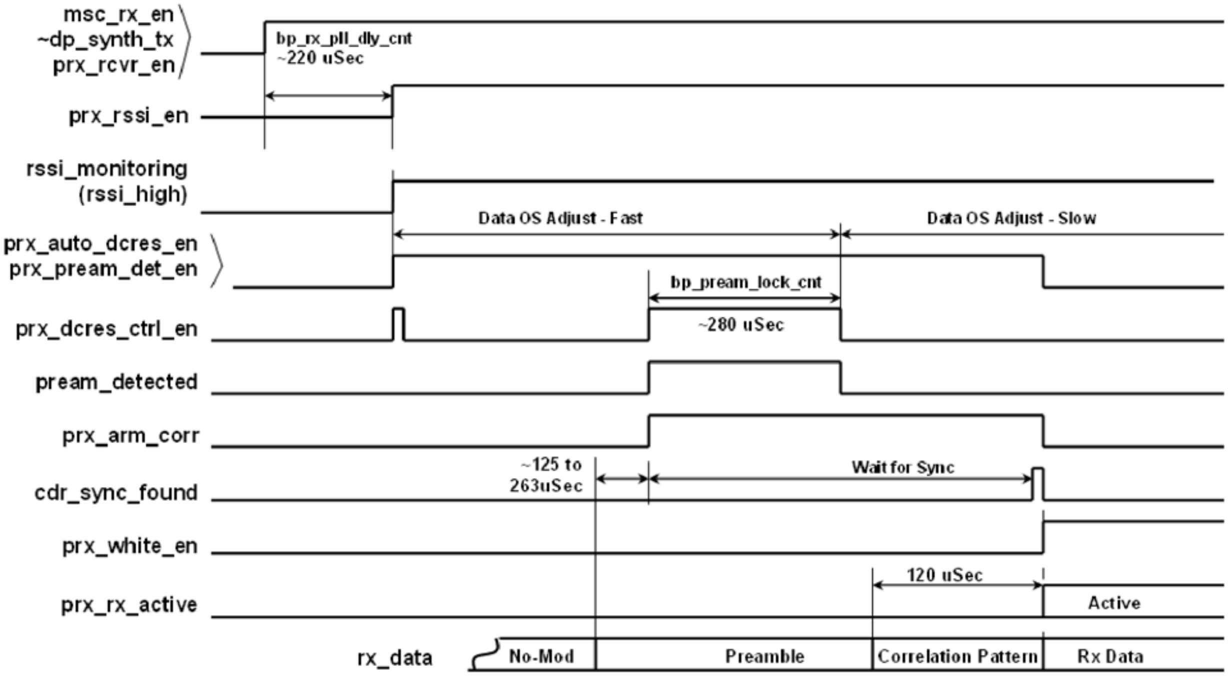
4.1.4.2 RX Timing Delays (Default Register Settings)

Figure 11, page 13, shows the timing for the typical RX startup using the default settings. The timing assumes that the PLL start-up delay is already complete, and that a transmit operation has recently completed. The **RX_PLL_DLY_CNT** is the settling time allowed for changing from the channel center frequency to the IF RX frequency. This delay is bypassed on device startup and if more than 300 μs has elapsed since the last TX. However, it is always present on the TX to RX turnaround.

Depending on the mode, there is a delay from when the RSSI-ADC block is turned on and when the RSSI is level is detected high. Because **ADC_RSSI_THRESH** is greater than zero (see recommended initialization value in Table 35, page 53), then the RSSI must be above the threshold before the receive process continues. If **ADC_RSSI_THRESH** is zero (not recommended), then the receive process commences immediately by either enabling the DC restore block, or enabling the preamble detector block, depending on the setting of *pream_det_mode*.

In preamble detect mode (default mode), the preamble detector detects the arrival of the preamble. At that time, it turns on the DC restore block. Once the frame synchronization pattern is detected, the receiver informs the MAC that the packet data is now valid.

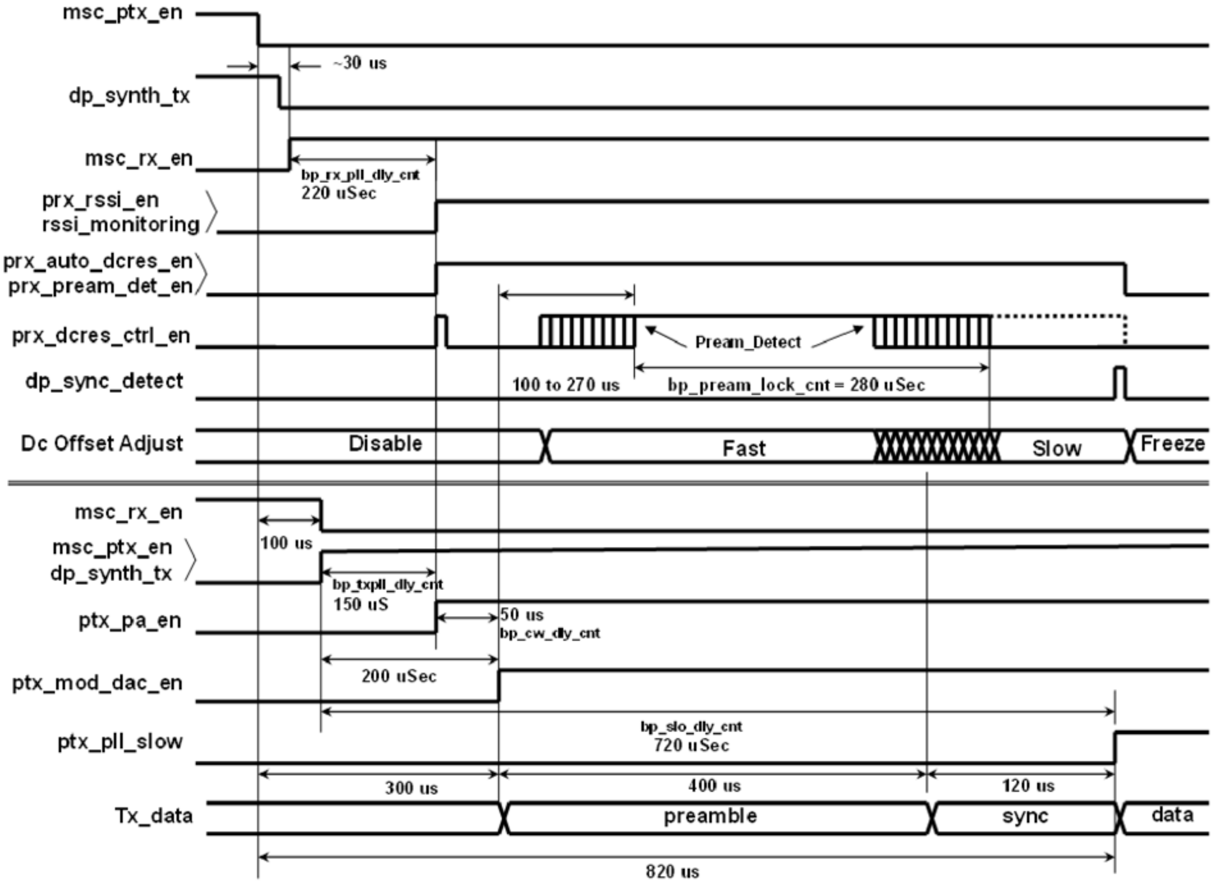
Figure 11 • **Default RX Start-Up Timing**



4.1.4.3 Optimized TX to RX Turnaround Timing with Preamble Detection

Figure 12, page 14, shows the TX to RX turnaround timing using preamble detect mode. The top section shows the device going from TX to RX, and the bottom portion shows the device going from RX to TX.

Figure 12 • **Optimized Turnaround in Preamble Detect Mode**



4.1.5 Packet Operations

4.1.5.1 Z-Star Node Operation, Transmit Transaction Sequence Mode

The node is typically set up to perform a single packet transmit transaction, starting with a CSMA operation. All the node MPU needs to do is write the payload into the node’s transmit buffer using a SPI packet write command, and then wait for the *cmd_done_irq* interrupt. While waiting, the MPU can be put in the sleep state.

When the interrupt is received, the node MPU needs to check for any errors. If a PLL lock error is present, then a VCO trim is required. Note that *cmd_done_irq* may not be the first interrupt to be asserted on a packet transaction when other interrupts are enabled. In particular, *rx_pkt_rdy_irq* and *rx_frm_pend_irq* can occur, if enabled, after the packet is transmitted and before the *cmd_done_irq* in both TX Packet and TX Data Request operations. In this case, the user needs to understand the associated timing and decide whether *rx_pkt_rdy_irq* and *rx_frm_pend_irq* should be enabled as interrupts or just used as status. In general, *cmd_done_irq* is the last interrupt to be asserted in a packet transaction, and the other interrupt/status can wait for *cmd_done_irq*.

If the *rx_pkt_rdy_irq* status/interrupt is received, then there is a packet in the RX buffer from the hub that needs to be read using a packet read operation. The length of the packet is in the SPI status when the interrupt is read.

If *cmd_fail_irq* and *rx_frm_pend_irq* status are both set, then the hub attempted to send the node a packet, and the packet was not received. In this case, the node MPU should send a SPI data request command to give the hub another opportunity to send the packet.

If *cmd_fail_irq* and *msc_rx_ack_fail* status are both set, then the node did not receive an acknowledgment for the transmitted packet from the hub after the programmed number of retries or because CSMA failed.

- The first option is to re-try the packet transmission.
- Next is to use a mode that gives more sensitivity with lower data rate and/or enabling FEC. In this case, it is necessary to have both devices setup in adaptive mode, with the hub in Follow receiver mode.
- If there is no response from the hub, then search for the hub on another channel using a SPI data request command.

4.1.5.1.1 Transmit Transaction Sequence

A packet transmission sequence is initiated as follows, after initialization and register setup as defined in Section 6 Setup and Initialization, page 35, for the node. It is assumed that the node has been put in a sleep state, where the main section of the chip is powered down. See Table 14, page 35.

1. Perform dummy read by performing a SPI read of address 0x000 (if the device is in sleep mode). This takes the device out of the SLEEP state and puts it in STANDBY.
   - Put MPU into sleep mode and wait for SPI ready interrupt on **IRQ** pin
2. Write payload to TX buffer using a SPI packet write command:
   - Put MPU into sleep mode and wait for interrupt on **IRQ** pin
   - After the interrupt, read registers **IRQ0**, **IRQ1**, and **IRQ2** in a single three-byte SPI read operation
3. If *cmd_done_irq* is received and *rx_frm_pend_irq* status is 1:
   - Write SPI data request command
   - Put MPU into sleep mode and wait for interrupt on **IRQ** pin
   - After the interrupt, read registers **IRQ0**, **IRQ1**, and **IRQ2** in a single three-byte SPI read operation
   - If *cmd_done_irq* is received and *rx_pkt_rdy_irq* status is 1, then go to Step 6
   - Else If *cmd_done_irq* is received and *msc_rx_null_data_frm*, then the command is complete; put the device into the SLEEP state via the SPI interface by writing 0xDE to SPIR_PWRDWN_REQ
   - Else If *cmd_done_irq* is received and *cmd_fail_irq* status is 1, go to Step 5

Note: If *rx_frm_pend_irq_en* is equal to 1, then this interrupt is asserted, and may occur prior to *cmd_done_irq*. Typically it is preferred to set *rx_frm_pend_irq_en* equal to 0 and to use *rx_frm_pend_irq* as status only.

4. Else If *cmd_done_irq* is received and *cmd_fail_irq* status is 0:
   - Mark TX packet as transmitted and remove it from TX queue.
   - Command is complete; put the device into the SLEEP state via the SPI interface by writing 0xDE to SPIR_PWRDWN_REQ
5. Else If *cmd_done_irq* is received and *cmd_fail_irq* status is 1:
   - Read MAC and receiver status and process error accordingly.
   - Command is complete; put the device into the SLEEP state via the SPI interface by writing 0xDE to SPIR_PWRDWN_REQ
6. Else If *cmd_done_irq* is received and *rx_pkt_rdy_irq* status is 1:
   - Save ***rxc_pkt_buf_len[8:0]*** from previous SPI interrupt status read (see Table 10, page 27)
   - Read packet using SPI packet read command
   - Perform any processing required by the received packet
   - Mark TX packet as transmitted and remove it from TX queue
   - Command is complete; put the device into the SLEEP state via the SPI interface by writing 0xDE to SPIR_PWRDWN_REQ
7. Else wait for MPU timeout. In this case, the TX packet was successful and acknowledged, and the hub successfully transmitted a packet back to the node.

Note: If *rx_pkt_rdy_irq_en* is equal to 1, then this interrupt is asserted, and may occur prior to *cmd_done_irq*. Typically it is preferred to set *rx_pkt_rdy_irq_en* equal to 0 and to use *rx_pkt_rdy_irq* as status only.

4.1.5.1.2 Transmit Data Request Transaction Sequence

1. Perform dummy read by performing a SPI read of address 0x000 (if the device is in sleep mode)
    - Put MPU into sleep mode and wait for SPI ready interrupt on **IRQ** pin
2. After the interrupt, write SPI data request command.
    - Put MPU into sleep mode and wait for interrupt on **IRQ** pin
    - After the interrupt, read registers **IRQ0**, **IRQ1**, and **IRQ2** in a single three-byte SPI read operation
3. If *cmd_done_irq* is received and *rx_pkt_rdy_irq* status is 1, then:
    - Save ***rxc_pkt_buf_len[8:0]*** from previous SPI interrupt status read (see Table 10, page 27).
    - Read packet using SPI packet read command
    - Perform any processing required by the received packet
    - Command is complete; put the device into the SLEEP state via the SPI interface by writing 0xDE to SPIR_PWRDWN_REQ
4. Else If *cmd_done_irq* is received and *msc_rx_null_data_frm* status is 1:
    - Command is complete; put the device into the SLEEP state via the SPI interface by writing 0xDE to SPIR_PWRDWN_REQ
5. Else If *cmd_done_irq* is received and *cmd_fail_irq* status is 1:
    - Read status and process error accordingly.
    - Command is complete; put the device into the SLEEP state via the SPI interface by writing 0xDE to SPIR_PWRDWN_REQ
6. Else wait for MPU timeout.

4.1.5.2 Z-Star Hub Operation, Receive Transaction Sequence Mode

The hub is typically set up to perform continuous receive operations. It only needs one SPI receive command to start it receiving packets. It waits for an *rx_pkt_rdy_irq* interrupt, and then reads the packet with a SPI packet read command. The acknowledgment packet is automatically sent to the node when a good packet is received.

When the hub receives a packet from the node, it should check its transmit queue for any pending packets to be sent to that node. If it has a packet for the node, and the packet received from the node was a non-data-request packet, then the hub MPU must set the *tx_pkt_frm_pend* and *tx_tn* bits to 1 for transmission in the acknowledgment packet, and then immediately write the packet into the transmit buffer with a packet write command.

This is a time critical operation. Both *tx_pkt_frm_pend* and *tx_tn* must be set before the acknowledgment packet header is transmitted, and the packet must be written to the TX buffer before the packet header for transmitted packet is written on this back-to-back TX-TX operation.

If a data request packet is received, then the hub only needs to write the packet to the TX buffer. No acknowledgment packet is sent. Once again, the packet must be in the buffer prior to the header being transmitted. This is because the length is contained in the header. If there is not enough time to write the entire packet, then the length must be written first to *tx_buf_len* and the SPI interface needs to be set to non-auto-length mode by setting *tx_auto_len_en* equal to 0.

When a packet is sent to the node, the hub should monitor for the *rx_ack_done_irq* interrupt and check *msc_rx_ack_fail* status to determine if the packet transmission was completed correctly.

4.1.5.2.1 Receive Transaction Sequence

A packet reception sequence is processed as follows when the packet is a data packet or data request packet. After initialization and register setup as defined in Section 6 Setup and Initialization, page 35, for the hub:

1. Set *rx_forever* equal to 1 on the hub
2. Write SPI receive command to begin receive process

3. Wait for interrupt on the **IRQ** pin
4. When the **IRQ** pin goes high, read registers **IRQ0**, **IRQ1**, and **IRQ2** in a single three-byte SPI read operation. Save ***rxc_pkt_buf_len*** from the SPI status (see Table 10, page 27).
5. If *rx_pkt_rdy_irq* is received and *msc_rx_data_frm* equals 1:
   - Read packet using SPI packet read command
   - Perform any processing required by the received packet
   - In MPU, check TX queue for a packet for that node.
   - If there is a packet in queue for that node, go to Step 8
   - Else (that is, if there is no packet for that node), done. Wait for another RX packet; go to Step 3
6. Else If *rx_pkt_rdy_irq* is received and *msc_rx_data_req_frm* equals 1:
   - Check the source ID from the SPI status received during the IRQ status read
   - In MPU, check TX queue for a packet for that node
   - If there is a packet for that node, go to Step 9
   - Else If there is no packet for that node:
     - Set *tx_null_frm* equal to 1
     - Set *tx_ack_frm_pend* equal 0 and *tx_tn* equal to 0
     - Wait for interrupt on **IRQ** pin
   - If *cmd_done_irq* is received and *tx_pkt_done_irq* status equals 1:
     - Set *tx_null_frm* equal to 0
   - Else, process error
   - Done; wait for another RX packet. go to Step 3
7. Else, process error
8. Set up for immediate packet TX to node after TX of acknowledgment packet:
   - Set *tx_ack_frm_pend* equal to 1 and *tx_tn* equal to 1
   - TX acknowledgment packet is automatically transmitted
9. Transmit packet to node
   - Write the packet to TX buffer using SPI packet write command
   - Wait for interrupt on **IRQ**
   - After the interrupt, read registers **IRQ0**, **IRQ1**, and **IRQ2** in a single three-byte SPI read operation
10. If *cmd_done_irq is* received and *rx_ack_done_irq* status is 1:
    - In MPU remove packet from TX queue.
    - Done; wait for another RX packet; go to Step 3
11. Else, TX packet failed
    - Mark TX attempt on TX queue; wait for another TX opportunity to same node
    - Done: wait for another RX packet; go to Step 3
12. Exit and wait for another packet; go to Step 3

Note: This process assumes that the device is a hub, and *rx_forever* is set to 1. With this setting, the hub returns to waiting in receive for another packet after each transaction is completed, regardless of whether or not the transaction passed or failed.

4.1.6 Adaptive Packet Reception

4.1.6.1 Auto Detection of Bit Rate

Auto bit rate detection (where *adapt_rate_en* is equal to 1) uses the receiver preamble detector to determine the bit rate. Three preamble detectors operate in parallel, each being set to one of the three rates. Refer to Table 5, page 18. The first to detect preamble determines the rate. Once the rate is selected, the rate controls the prefilter bandwidth, the AFC/DC restore time and bandwidth, and the clock and data recovery rates.

Once the preamble is detected, the corresponding rate is latched as *prx_rx_rate*. When *adapt_rate_en* is set to 0, *rx_rate* controls the receiver behavior. When *adapt_rate_en* is set to 1, the RX rate is controlled by *prx_rx_rate* instead of *rx_rate* in the PHY RX controller.

4.1.6.2 Auto FEC Detection

When *adapt_fec_en* is set to 1, the receiver frame synchronization correlator searches for both inverted and noninverted frame sync patterns. If a normal frame synchronization pattern is found, then the FEC decoder in the receiver are by-passed. If an inverted frame synchronization pattern is found, then the receiver FEC decoder is enabled and used for error correction of the RX data.

The MAC enables FEC on the MAC receiver if an inverted sync was received. If *adapt_fec_en* is set to 0, it uses *hmd_dec_en* to enable FEC on the MAC receiver.

If *tx_follow_rx_fec* is set to 1, the transmitter responds to a received packet with the same FEC mode (on/off) that it received. The bit *cdr_inverted_sync* indicates that the previous frame sync was inverted. When *tx_follow_rx_fec* is set to 1, the transmitter uses the *cdr_inverted_sync* bit to determine whether FEC should be enabled and whether the frame synchronization pattern should be inverted.

The polarity of the transmitter frame synchronization pattern and the FEC enable state is shown in Table 4, page 18.

Table 4 • **Settings for TX Frame Sync Polarity and FEC Enable**

adapt_fec_en	0	0	1	1	1	1
tx_follow_rx_fec	X	X	0	0	1	1
hme_enc_en	1	0	1	0	X	X
cdr_inverted_sync	X	X	X	X	1	0
Frame synchronization polarity	Pos	Pos	Pos	Neg	Pos	Neg
FEC enable	1	0	1	0	1	0

4.1.6.3 Transmitted Preamble

The preamble bit pattern is always 00110011, which is expressed in bit periods and not symbols because the transmitter deals only with bits. (For example, the preamble symbol pattern for a bit rate of 100 kbit/s is 00110011 in bits, but it is 0000111100001111 in symbols.)

Table 5 • **TX Preamble Pattern**

tx_rate	**Preamble Pattern In Bits**	**Bit Rate**
00	00110011	200 kbit/s
01	00110011	100 kbit/s
10	00110011	50 kbit/s

4.1.6.4 Transmitted FEC and Frame Sync Pattern

The TX hamming enable is sourced from *hme_enc_en* when *tx_follow_rx_fec* is 0, or from the receiver *cdr_inverted_sync* if *tx_follow_rx_fec* is 1 to automatically transmit in the same FEC mode as the packet received. In the latter case, enabling FEC causes the frame synchronization pattern to be transmitted inverted. See Table 6, page 19.

Table 6 • **Enable for TX FEC Encoding**

tx_follow_rx_fec	*hme_enc_en*	*cdr_inverted_sync*	**FEC Encoding**
0	0	X	OFF
0	1	X	ON
1	X	0	OFF
1	X	1	ON

If *adapt_rate_en* is set to 1, then the transmitted frame sync pattern is inverted if FEC is enabled and noninverted if FEC is disabled. If *adapt_rate_en* is set to 0, then the transmitted frame sync pattern is never inverted.

4.1.6.5 Receiver FEC Decoding

In the receiver, the hamming FEC decoder block is enabled based on *hmd_dec_en*, *adapt_fec_en* and *cdr_inverted_sync*, according to Table 7, page 19. If enabled, the 12-bit symbols are then decoded using a code (12,8) hamming decoding. This decoding process corrects any one-bit error in the 12-bit symbol, with no indication that an error was corrected. The output of the decoding is ***rxc_hmd_data[7:0]***, which is then processed by the rest of the receiver.

Table 7 • **Receiver Hamming Decoder Enable**

adapt_fec_en	*hmd_dec_en*	*cdr_inverted_sync*	**FEC Decode**
0	1	X	Yes
0	0	X	No
1	X	1	Yes
1	X	0	No

4.1.7 Packet Reception

The receiver searches for the preamble pattern before enabling the search for a valid frame sync.

Preamble detection detects a valid preamble sequence and uses this detection to enable the AFC/DC settling function. The preamble detector delays starting the AFC and DC-restore settling function until a valid preamble is detected, helping to ensure that the AFC/DC settling function is performed on a valid signal and not on interference or other RF transmissions. Once preamble detection has occurred, the receiver starts searching for frame sync. In the event that frame sync is not detected, usually due to too many bit errors, the use of a timer is recommended to abort searching for frame sync and return to searching for preamble. This timer begins counting after preamble detection has occurred. The default timeout period is 15 bytes after preamble is detected, which is the recommended value to reduce the possibility of a false preamble detect due to a preamble detection that was the result of noise and not a valid preamble.

4.2 Channel Monitoring

Typically, it is necessary to check that a channel is not busy before transmitting a packet. CSMA is an algorithm used to implement this channel monitoring. A Carrier Sense Multiple Access (CSMA) protocol is used to avoid collisions with other transmissions or interference on a specified channel.

4.2.1 CSMA Threshold

In CSMA, an RSSI threshold is set, *adc_csma_thresh*, which defines the minimum RSSI level at which the device is allowed to transmit. If the RSSI level is greater than *adc_csma_thresh*, then the device must wait until the level goes below *adc_csma_thresh*. To set the RSSI threshold for use with CSMA, a RSSI measurement must be made (see 4.2.2 RSSI Measurement, page 20) and then set the *adc_csma_thresh* register to 24 counts above the RSSI result found in *adc_avg*.

4.2.2 RSSI Measurement

The procedure in Table 8, page 20, describes the programming steps required in performing an RSSI measurement. Ensure all calibrations have been performed (refer to Section 7 Calibrations, page 40) prior to making an RSSI measurement.

Table 8 • **Procedure for Manual RSSI Measurement**

1. Write **MAC_CTRL** = 0x00.	Disables the MAC.
2. If the *adc_done_irq* interrupt is used, write **IRQ_EN1[4]** = 1.	Setting the *adc_avg_done_irq_en* bit enables the *adc_done_irq* interrupt.
3. Write **MAN_TEST[2]** = 0	Clearing the *rfmac_synth_tx* bit selects RX mode.
4. Write **ADC_MUX_IN_SEL** = 0x04.	Writing this value to the *adc_mux_in_sel* bits selects the RSSI as the input to the ADC.
5. Write **ADC_POW_N_CONV[3:0]** = 4’b0011.	Bits [3:0] set the sample size for the number of sequential RSSI conversions to eight (recommended value).
6. Write **MAN_GLOBAL_EN** = 0x02.	Setting the *rfmac_rcvr_en* bit manually enables the receiver.
7. Wait 2ms for the PLL to settle.	
8. Write **ADC_CONV_START[1]** = 1.	Bit [1] initiates the ADC conversions. Two results are obtained from this sample size: an average RSSI and a maximum RSSI.
9. Wait for interrupt or time it takes for conversion to complete.	Each ADC conversion takes approximately 10 µs. After all eight conversions are complete, bit [1] of **ADC_CONV_START** is cleared internally.
10.Read **ADC_AVG** and **ADC_MAX**.	These result registers give the average RSSI over eight samples and the maximum over eight samples, respectively.
11.Clean up after procedure: a.Restore **MAC_CTRL** value. b. Restore **MAN_TEST** value. c. Restore **ADC_MUX_IN_SEL** value. d. Restore **ADC_POW_N_CONV** value. e. Restore ***MAN_GLOBAL_EN*** value. f. Restore **IRQ_EN1** value, if used.	Be sure to restore these registers to the original values that were stored before this procedure began.

4.3 State Machine Diagrams

This section describes the operation of the communication protocol with specific reference to the sequence of operations performed in various modes. State machine diagrams depict the control actions performed internally. These diagrams are useful for understanding the following:

- Differences in operation of hub and node
- Packet transaction sequences

These overview descriptions identify only the major operational phases and states.

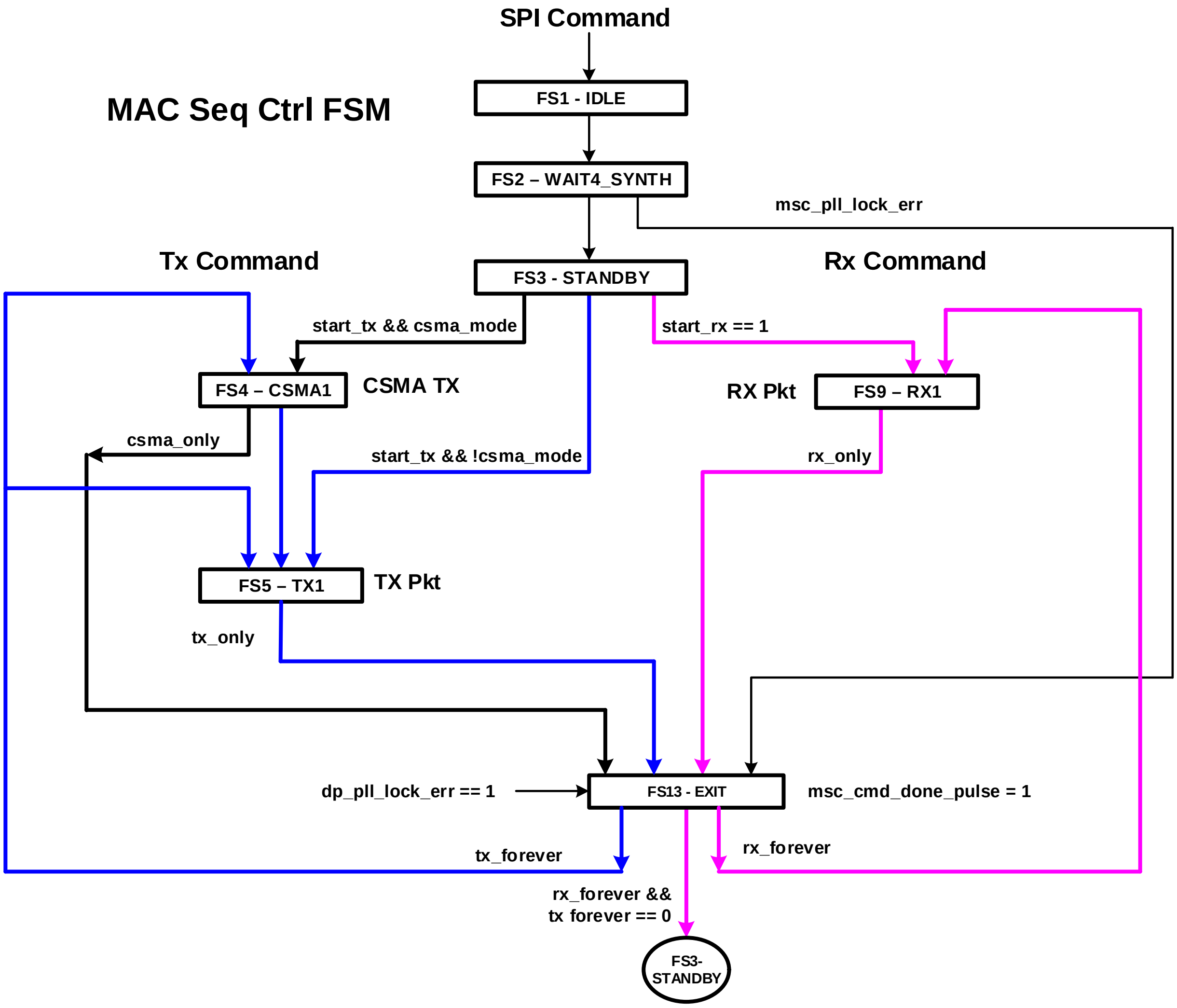
4.3.1 Single Packet Operation

Figure 13, page 21, shows the MAC state machine operation for single packet operation. In this mode, a single packet is transmitted or received, and then the operation completes and the SPI command is cleared. To enable this mode, *tx_only* and *rx_only* are set to 1.

In this mode, no acknowledgment is sent on packet reception, nor is there a switch to receive for an acknowledgment after transmit. Likewise, there is no retry on a failed acknowledgment.

A CSMA may optionally be performed prior to transmitting a packet by setting *csma_mode* equal to 1 and *csma_only* equal to 0. To perform a CSMA only and then exit before transmit, set *csma_mode* equal to 1 and *csma_only* equal to 1.

Figure 13 • **Simplified Single Packet Mode State Machine**



Note: Definitions:
- ***start_tx*** is given by EITHER: (a) a SPI Packet Write command with *auto_tx_cmd* equal to 1 (where 1 is default) OR (b) a SPI Start Transmit Command OR (3) a SPI Transmit "Data-Request" Command. Refer to the SPI command encoding information in Figure 16, page 26.
- ***start_rx*** is given by a SPI Receive command.

4.3.2 Transaction Sequence Operation

In transaction sequence mode, packet operations are completed per Figure 14, page 23, and Figure 15, page 24.

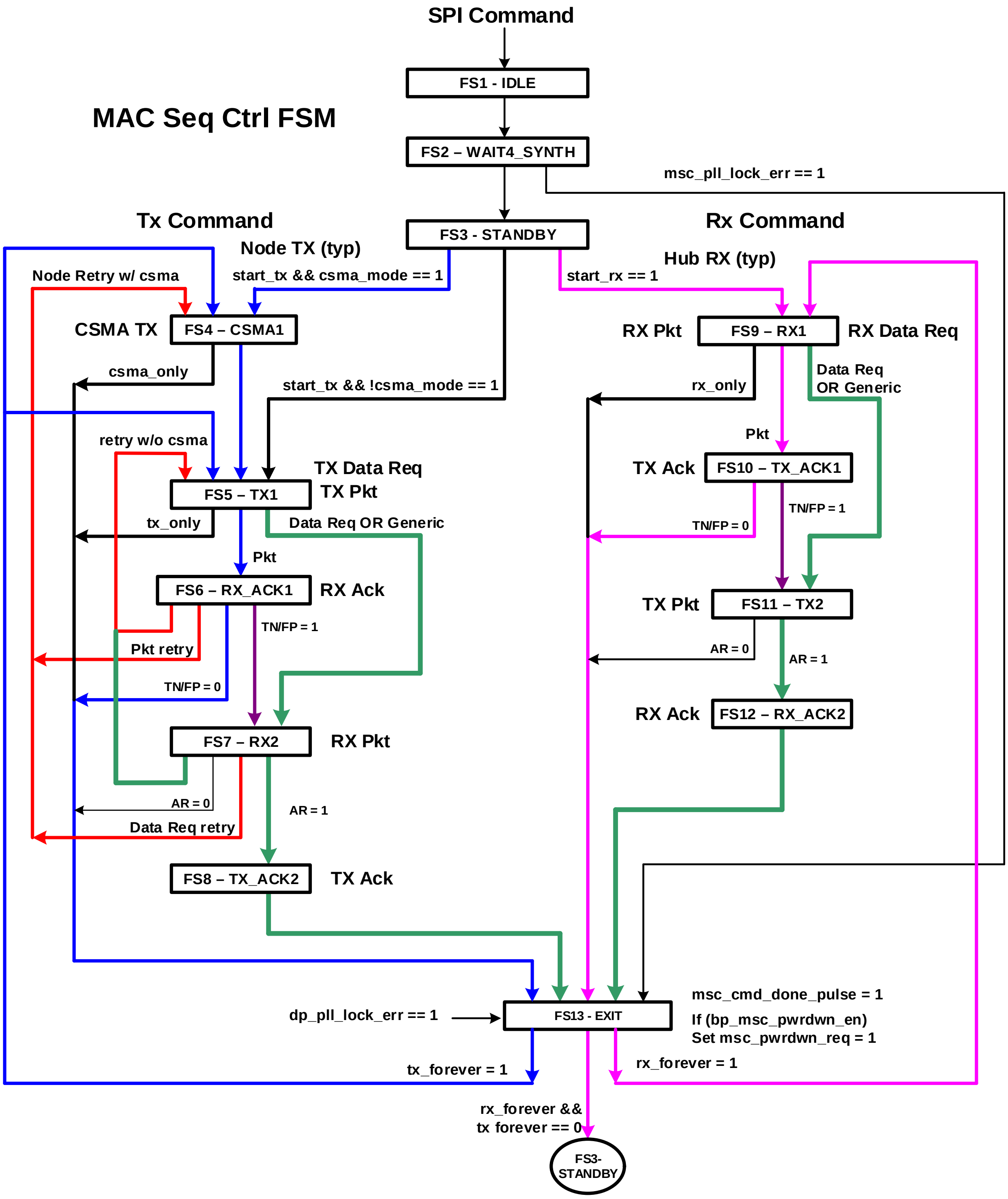
The hub is placed in receive Forever mode, and the node receives transmit commands from its host. In Z-Star packet mode, acknowledgment packets are automatically sent and received, as are other operations such as data requests and transmit immediate operations from the hub to the node. If a transmission from the node Fails, then an automatic retry of the previous packet transmission is performed up to the programmed retry count.

In user packet mode and raw packet mode, the setup is similar. The node transmits to the hub. When the hub receives a good packet, it automatically turns around to transmit a response, which must be immediately loaded into the transmit buffer. After the node transmits, it also turns-around to receive a response packet. If the expected packet is not received correctly, then the node automatically retransmits the previous packet up to the retry count.

4.3.2.1 Z-Star Packet Operation

Figure 14, page 23, shows the Z-Star operating state machine used in transaction sequence mode.

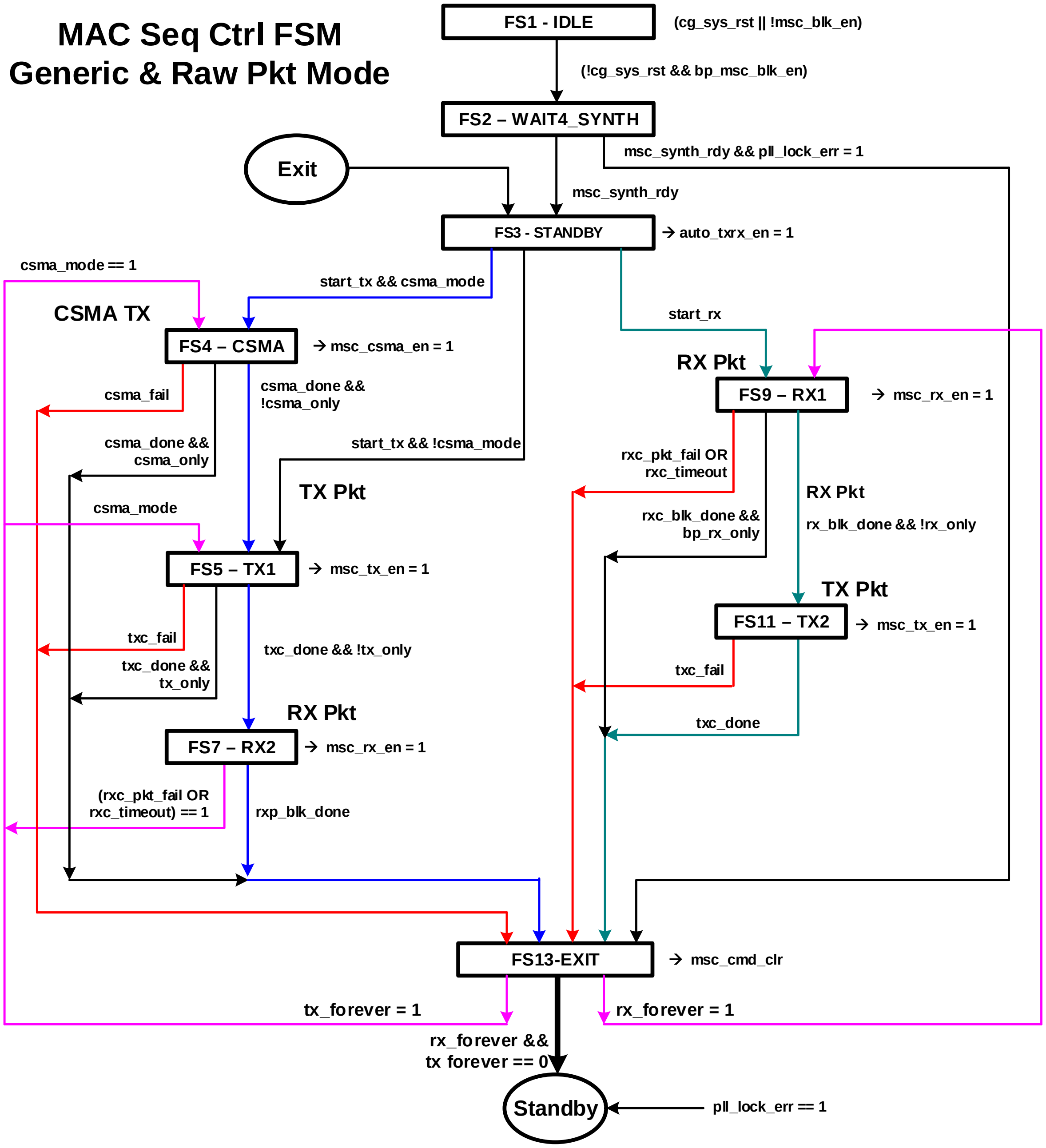
Figure 14 • **Simplified MAC Sequencer State Machine**



Note: Definitions:
- ***start_tx*** is given by EITHER: (a) a SPI Packet Write command with *auto_tx_cmd* equal to 1 (where 1 is default) OR (b) a SPI Start Transmit Command OR (3) a SPI Transmit "Data-Request" Command. Refer to the SPI command encoding information in Figure 16, page 26.
- ***start_rx*** is given by a SPI Receive command.

4.3.2.2 User Packet Operation and Raw Packet Operation

Figure 15 • **State Machine for User Packet Mode and Raw Packet Mode**



Note: Definitions:

- ***start_tx*** is given by **either**: (a) a SPI Packet Write command with *auto_tx_cmd* equal to 1 (where 1 is default) **or** (b) a SPI Start Transmit Command **or** (3) a SPI Transmit "Data-Request" Command. Refer to the SPI command encoding information in Figure 16, page 26.
- ***start_rx*** is given by a SPI Receive command.
- ***txc_fail*** is given by *msc_tx_pkt_fail*.

5 Application Interface

5.1 Introduction

The ZL70550 has a highly versatile application interface, the general features of which are outlined in this chapter.

Two fundamental points about the ZL70550 design are important to this discussion.

- Hardware state machine

  The ZL70550 is designed as a memory-mapped, hardware state machine. The state machine architecture means that all the digital functions described in this User's Guide are built into the hardware. They are not reconfigurable in software.
- Memory-mapped device

  In most respects the ZL70550 can be controlled by the application program as a memory-mapped peripheral. The ZL70550 memory map is divided into eight-bit registers, and most interactions with the device are by register writes and reads.

5.1.1 Support for Peripheral Access and Control

The ZL70550 supports interrupt-driven peripheral access and control. In this method, the peripheral device initiates the servicing process by asserting an interrupt to the application processor. Interrupt-driven systems may be more power efficient because the application processor does not have to run timers and poll unnecessarily.

For access to its registers, the ZL70550 provides a serial peripheral interface conforming to industry-wide SPI standards. The ZL70550 has an IRQ output that can be asserted by one or more interrupt sources, which can be individually disabled or enabled (masked or unmasked).

Programmable outputs are also available for use. Internal signals such as various interrupt sources may be selected to output to the general-purpose I/O pins as described in Section 5.5 GPIO Pins, page 32.

5.2 Serial Peripheral Interface

This section describes the Serial Peripheral Interface (SPI), which is used to read or write to registers in the ZL70550 memory map, including the TX and RX buffers.

5.2.1 SPI Principles

SPI is an industry standard for in-system communication. Using a synchronous bus, data is transmitted and received in eight-bit words.

Multiple devices may connect to the SPI bus, but only one acts as the bus master. The ZL70550 always operates as a SPI slave device, so the user application must be programmed as a master.

Four signal lines are used for SPI communication:

- **SPI_CLK**, the synchronizing clock signal, sent to the ZL70550 RF transceiver by the master device.
- **SPI_MOSI**, SPI bus data input.
- **SPI_MISO**, SPI bus data output.
- **SPI_SEL_B**, the SPI bus select signal. Note that this signal is active low.

5.2.2 SPI Command Set

The SPI command set is shown in Table 9, page 26. The MOSI line shows the bit stream that is output to the host CPU. The MISO line shows the data output from the ZL70550 SPI slave interface. In all cases, one to three status bytes are output on MISO at the beginning of every SPI operation. See Table 10, page 27, for information on the SPI status.

Each SPI operation is initiated with a two-bit to eight-bit sequence, as defined in Table 9, page 26. Read and write operations occur at specified address. See Table 11, page 30, and Chapter 8 Registers, page 52.

Table 9 • **SPI Command Set**

SPI Command	Command Format	No. Status Bytes
Short-address write	MOSI: <2'b11,Address><wr_data> …. <wr_data> MISO: <Stat_0><0x00>….<0x00>	1
Short-address read	MOSI: <2'b01,Address><0x00>….<0x00> MISO: <Stat_0><rd_data> …. <rd_data>	1
Long-address write	MOSI: <4'b1010,Address><Address><wr_data>….<wr_data> MISO: <Stat_0><0x00>….<0x00>	1
Long-address read	MOSI: <4'b0010,Address><Address> MISO: <Stat_0><Stat_1><Stat_2><rd_data>….<rd_data>	3
Packet write	MOSI: <0x88><wr_data>….<wr_data> MISO: <Stat_0><0x00>….<0x00>	1
Packet read	MOSI: <0x88><0x00>….<0x00> MISO: <Stat_0><Stat_1><rd_data>….<rd_data>	2
Transmit command	MOSI: <0x89> MISO: <Stat_0>	1
Transmit data request command	MOSI: <0x8C> MISO: <Stat_0>	1
Receive command	MOSI: <0x8D> MISO: <Stat_0>	1
Abort command	MOSI: <0x8F> MISO: <Stat_0>	1

5.2.3 SPI Command Encoding

The SPI command encoding is shown in Figure 16, page 26.

Figure 16 • **SPI Command Encoding**

```
11AA  AAAA  | DDDD DDDD | {DDDD DDDD} - Short Address Write
1010  AAAA  | AAAA AAAA |  DDDD DDDD | {DDDD DDDD} - Long Write
1000  1000  | DDDD DDDD | {DDDD DDDD} – Packet Write
1000  1001  | Start Transmit Command
1000  1100  | Transmit “Data-Request” Command
1000  1101  | Start Receiver Command – Enable Receiver
1000  1111  | Abort Command

01AA  AAAA  | RRRR RRRR | {RRRR RRRR} – Immediate Address Short Read
0010  AAAA  | AAAA AAAA |   XXXX XXXX  | RRRR RRRR | {RRRR RRRR} – Long Read
0000  1000  | XXXX  XXXX  |  RRRR RRRR | {RRRR RRRR} – Packet Read
```

Note:

1. A = Address bit, on MOSI
2. D = Write data bit, on MOSI
3. R = Read data bit, on MISO
4. X = Dead bus cycle, used for status on MISO

5.2.4 SPI Status

On all SPI operations, one to three bytes of status information is returned to the host on MISO, as shown in Table 10, page 27. The number of status bytes returned per SPI operation is shown in Table 9, page 26. The contents of status byte 2 depend on the setting of *hub_node_n*, which is set to 1 for hub configuration and to 0 for node configuration.

Table 10 • **SPI Status Returned on MISO**

Status Byte	Bit Position	Signal Name	Description
0	7	***~cg_sys_spi_rst***	Main chip reset, from ***cg_sys_rst***. When high, indicates that VDDD and SPI are ready
0	6	***spis_bad_cmd_stat***	Bad SPI command on last command
0	5	***spis_radio_rdy***	Radio ready for transmit or receive packet command
0	4	*rxc_pkt_pass*	No CRC or address error on last RX packet
0	3	*rxc_ack_req*	Acknowledge request bit set on last RX packet
0	2	*rxc_brdcst_match*	RX broadcast packet
0	1	*rxc_addr_mode*	RX packet address mode
0	0	***rxc_pkt_buf_len[8]***	Bit [8] (MSB) of RX packet length in RX buffer. See Note 1.
1	7:0	***rxc_pkt_buf_len[7:0]***	Bits [7:0] of RX packet length in RX buffer. See Note 1.
2 (hub)	7:0	*rxc_source_sid*	RX packet short ID of source
2 (node)	7	*msc_rx_dup_frm*	Last RX packet was a duplicate packet
2 (node)	6	*rxc_frm_pend*	Frame pending bit set on last RX packet
2 (node)	5:3	*rxc_frm_stype*	RX packet frame subtype
2 (node)	2:0	*rxc_frm_type*	RX packet frame type

1. The value of ***rxc_pkt_buf_len[8:0]*** is the number of bytes the ZL70550 puts in its RX buffer for the user to read out.
- If *wr_rx_payld_only* is set to 1 then the ZL70550 puts only the packet's payload in its RX buffer and ***rxc_pkt_buf_len[8:0]*** is the payload length *rxc_pld_len*.
- If *wr_rx_payld_only* is set to 0, then the ZL70550 puts the whole packet in its RX buffer, and ***rxc_pkt_buf_len[8:0]*** is the length of the whole packet and this is the value in *rxc_frm_len*.

5.3 SPI Commands

Read and write operations are performed with either short addresses or long addresses, depending on where in the memory map the operation is conducted. In all cases, there is a command field of two to four bits, followed by an address, followed by data. Multiple addresses may be accessed in a single command, with the address being automatically incremented from the starting address on consecutive bytes.

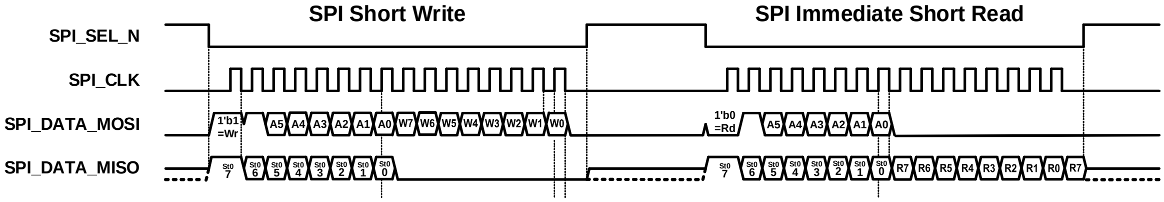
5.3.1 Short-Address Read and Write

Short addressing is used to access addresses 0x000 to 0x007. These operations occur in the SPI clock domain and do not require the crystal oscillator or system bus to be operational. Therefore no system bus synchronization is required. The timing for these operations is shown in Figure 17, page 28.

A write starts with 2’b11 followed by six bits of address. This is followed by one or more bytes to be written to consecutive address locations, starting with the first address.

A read starts with 2’b01 followed by six bits of address. This is followed by one or more bytes read from consecutive address locations, starting with the first address.

Figure 17 • **Short-Address Read-Write Timing**


SPI Short Write
SPI Immediate Short Read
SPI_SEL_N
SPI_CLK
SPI_DATA_MOSI
SPI_DATA_MISO
1'b1 =Wr
A5 A4 A3 A2 A1 A0 W7 W6 W5 W4 W3 W2 W1 W0
St0 7 St0 6 St0 5 St0 4 St0 3 St0 2 St0 1 St0 0
1'b0 =Rd
A5 A4 A3 A2 A1 A0
St0 7 St0 6 St0 5 St0 4 St0 3 St0 2 St0 1 St0 0 R7 R6 R5 R4 R3 R2 R1 R0 R7

5.3.2 Long-Address Read and Write

Long addressing is used to access all addresses on the system bus. See Table 11, page 30. The transmit and receive buffers may also be accessed with long-address read and write operations, but typically they are accessed with packet write and packet read operations.

The long-address operations use a 12-bit address. Write operations occur on consecutive byte cycles, with no dead byte periods. Because of the system bus synchronization requirements, a dead byte cycle in inserted after the address on read operations. In all cases, status in returned on the first two or three bytes on the MISO line, as shown in Table 9, page 26.

Figure 18, page 28, shows the timing and bit sequence for a long-address read operation, where two data bytes are read back on the SPI bus. There is a one byte delay between the address and the first data byte, in which SPI Status 2 is inserted. The read operations must perform a look-ahead in order to have the data ready for the SPI interface. Because of this, the SPI interface always reads two additional locations on the system bus after the last byte needed for the read operation. Notice that there are four bus requests for the two-byte read operation. The de-assertion of **SPI_SEL_B** high terminates the operation.

Figure 18 • **Long-Address Read Timing**

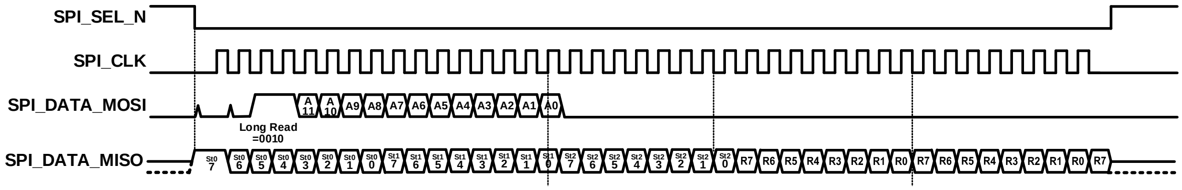
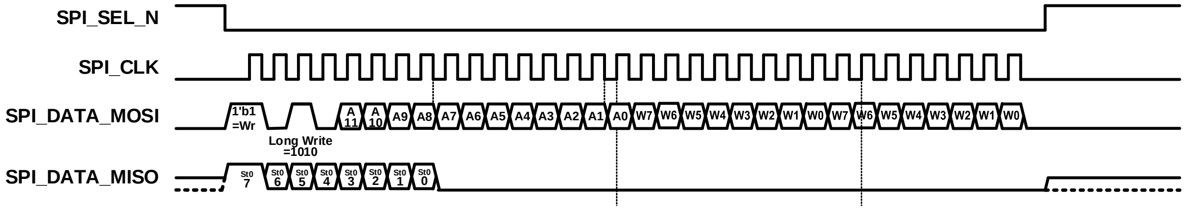

SPI_SEL_N
SPI_CLK
SPI_DATA_MOSI
SPI_DATA_MISO
A 11 A 10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
Long Read =0010
St0 7 St0 6 St0 5 St0 4 St0 3 St0 2 St0 1 St0 0 St1 7 St1 6 St1 5 St1 4 St1 3 St1 2 St1 1 St1 0 St2 7 St2 6 St2 5 St2 4 St2 3 St2 2 St2 1 St2 0 R7 R6 R5 R4 R3 R2 R1 R0 R7 R6 R5 R4 R3 R2 R1 R0 R7

Figure 19, page 28, shows the timing and bit sequence for the write operation, where two bytes are shown written on the system bus. In this case, there is no dead byte cycle between the last address bit and the first data byte to be written. The de-assertion of **SPI_SEL_B** high terminates the operation.

Figure 19 • **Long-Address Write Timing**


SPI_SEL_N
SPI_CLK
SPI_DATA_MOSI
SPI_DATA_MISO
1'b1 =Wr
Long Write =1010
A 11 A 10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0 W7 W6 W5 W4 W3 W2 W1 W0 W7 W6 W5 W4 W3 W2 W1 W0
St0 7 St0 6 St0 5 St0 4 St0 3 St0 2 St0 1 St0 0

5.3.3 Packet Write

Packet write operations are an efficient means of writing packets and starting the packet transaction. This operation is similar to the long-address write operation, except that no address is in the SPI command sequence. The packet write operation starts with the 0x88 command. In addition to writing the packet into the transmit buffer, this operation optionally performs three other operations:

- Calculates the TX buffer length, and writes it to *tx_buf_len*
- Increments the TX frame sequence number and writes it to *tx_fcs_len*
- Asserts the SPI transmit command *spis_tx_cmd* to the MAC to transmit the packet (see 5.3.5.1 Transmit Command, page 29.

If auto-header generation is enabled by setting *tx_auto_hdr* equal to 1, then the CPU only needs to write the payload into the TX buffer using the packet write operation, and the entire packet is automatically generated with preamble, frame sync pattern, PHY header, MAC header, payload, and CRC.

5.3.4 Packet Read

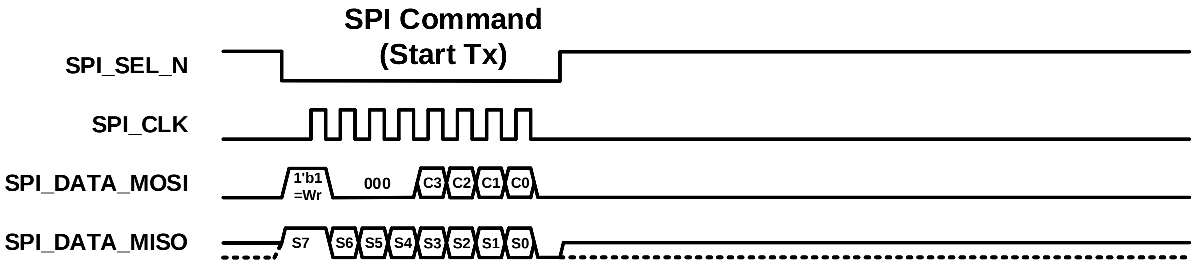
This operation is similar to the long-address read operation, except that no address is in the SPI command sequence. Packet read operation reads the packet currently stored in the RX packet buffer. Information in the status can be used to determine the state of the packet, including its length. This command does not turn on the receiver to receive a packet. This must be done with a SPI receive command, or in conjunction with a SPI TX command or SPI TX request command when operating in transaction sequence mode.

Depending on the setting of *wr_rx_payld_only*, either the entire packet or just the payload is written to the RX buffer. All status for a received packet is stored in the status registers until the next packet is received. The SPI status field ***rxc_pkt_buf_len[8:0]*** (see Table 10, page 27) gives the length of the RX data in the RX buffer.

5.3.5 Other SPI Commands

Figure 20, page 29, shows the timing for a typical one byte SPI command, in this case a TX command. The command is synchronized to the system clock ***cg_sys_clk*** before asserting *spis_tx_cmd* to the MAC. The *spis_tx_cmd* is cleared by the MAC when the command is complete.

Figure 20 • **Command Timing**



5.3.5.1 Transmit Command

The SPI transmit command *spis_tx_cmd* goes high either with a SPI Write Packet command (if auto_tx_cmd is 1) or with a SPI Transmit Command. The *spis_tx_cmd* command causes a SPI TX command to the MAC, which in turn begins a transmit packet sequence that may include CSMA, packet transmit, and packet receive operations as shown in the state machine diagrams (see Figure 13, page 21, Figure 14, page 23, and Figure 15, page 24). In this case, for the initial transmitted packet, the header and packet length is taken from the always-on registers, and the payload is taken from the TX buffer. If *tx_auto_hdr* is equal to 0, then the entire packet with the MAC header is taken from the TX buffer.

This command is useful for issuing a retransmission of the last packet transmitted, or for delaying transmission after a packet is written to the TX buffer.

5.3.5.2 Transmit Data Request Command

This SPI command *spis_tx_req_cmd* goes high with a SPI Transmit "Data Request" Command. The *spis_tx_req_cmd* causes the transmission of the data-request packet in Z-Star packet mode, which in turn begins a transmit packet sequence that may include CSMA, packet transmit, and packet receive operations as shown in the state machine diagrams (see Figure 13, page 21, Figure 14, page 23, and Figure 15, page 24). Typically, this is done from the node and requires no changes to the contents of the always-on registers. For the transmitted data request packet, the frame sequence number is set to the expected frame sequence number of the next RX packet, and the payload length is set to zero. Also, the AR and FP bits are also set to 0, and the frame type / frame subtype are set to data request type packet.

5.3.5.3 Receive Command

The SPI receive command *spis_rx_cmd* goes high with a SPI Receive Command. The *spis_rx_cmd* command is used to turn on the receiver to begin a receive packet sequence that may include both packet transmit and packet receive operations as shown in the state machine diagrams (see Figure 13, page 21, Figure 14, page 23, and Figure 15, page 24). It takes the MAC to state RX1, per Figure 14, page 23. This command is required in order to initiate a receive operation, and the receive operation must complete before a packet can be read from the RX packet buffer.

5.3.5.4 Abort Command

The SPI abort command is used to terminate an active command. This is most commonly used after a receive command has been issued when *rx_forever* is equal to 1. In this case, the RX command is not cleared without an abort command or a MAC disable/reset or a SLEEP condition.

5.4 Registers, Interrupts, and Memory Map

The user application controls most digital functions in the ZL70550 by accessing designated memory addresses called registers. Each register represents a designated address within the ZL70550 memory map.

5.4.1 Registers

On the SPI bus, each register is accessed using its 6-bit or 12-bit address, depending on whether short- or long-address mode is used for that register. Each register contains up to one byte of data, and the "Default" column of the memory map (refer to Section 9.3 SPI Local Registers, page 61, Section 9.4 System Bus Control and Status Registers, page 63, and Section 9.5 Always-On System Bus Control and Status Registers, page 75) identifies how many bits are significant for each register. These are always the least significant bits. Over the SPI bus, the MSB of each byte is always sent first, the LSB last. Read and write access is described in Section 5.3 SPI Commands, page 27.

In this User's Guide, individual bits within a register are typically referenced using the "Field Name" shown in the tables in Section 8 Registers, page 52, but may also be referenced using the bit number appended to the register name. In the latter case, a bit range within a register is identified in brackets and the number sequence of a bit range is always written as [MSB:LSB]. For example, the RX FCS length is stored in bits [2], [1], and [0] of register **RX_CTRL1** and may be referenced as **RX_CTRL1[2:0]** or as *rx_fcs_len*.

Registers are of two types: volatile and always-on registers. The contents of the volatile registers located in the VDDD power domain are lost every time the device goes into the SLEEP state. The always-on registers are located in the VSUP power domain and retain their values when the device goes into the SLEEP state.

5.4.2 Basic Memory Map

Table 11, page 30, summarizes the memory map of the ZL70550. Additional details can be found in Chapter 8 Registers, page 52. As indicated in the table, there are always-on registers whose values are always preserved as long as power is applied to VSUP, even while the chip is in the SLEEP state.

Table 11 • **Basic Memory Map**

Starting Address	Ending Address	Addressing Mode	Always On	Description
0x000	0x007	Short	No	SPI local registers, not on system bus
0x100	0x14B	Long	No	System bus control and status registers
0x200	0x29F	Long	Yes	System bus control and status registers
0x400	0x5FF	Long	No	System bus transmit buffer (SRAM)
0x600	0x7FF	Long	No	System bus receive buffer (SRAM)

5.4.3 Interrupts

The naming convention for interrupt registers is *interruptname_irq* (for example, *cmd_done_irq*).

Interrupts are signaled by the ZL70550 asserting its IRQ (Interrupt Request) output.

5.4.4 Interrupt Controller

There is only one interrupt pin at the top level of the ZL70550 chip called **IRQ**, and the interrupt sources are determined by writing to the enable registers **IRQ_EN0**, **IRQ_EN1**, and **IRQ_EN2**. More than one interrupt source per register can be enabled. In this case, when the **IRQ** pin is set high, it is necessary to read the interrupt registers **IRQ0**, **IRQ1**, and **IRQ2** (with an SPI bus read command) to determine what the source(s) is.

When the corresponding interrupts are enabled, the outputs of the interrupt registers **IRQ0**, **IRQ1**, and **IRQ2** create an interrupt on the **IRQ** pin (level high). If an interrupt is not enabled, the interrupt pulse in the interrupt registers **IRQ0**, **IRQ1**, and **IRQ2** can still be latched but that does not generate an interrupt on the **IRQ** pin.

If an interrupt is enabled and it is set, the **IRQ** pin is set high until it is lowered either by reading the respective interrupt register or by writing to the appropriate enable register to disable the interrupt.

The **STATUS** register returns the status as defined in the memory map when it is read, but a status can never generate an interrupt on the **IRQ** pin.

The **STATUS** register is read-only and registers **IRQ0**, **IRQ1**, and **IRQ2** are clear-on-read (CoR). Because of the prefetch behavior of the system bus, if just **IRQ0** is read, then **IRQ1** and **IRQ2** are also cleared. It is best to perform a three-byte read, starting with **IRQ0**, to read the interrupt status and clear the registers. This guarantees that no interrupt status is missed.

See Table 12, page 31, for a list of the individual ZL70550 interrupts.

Table 12 • **ZL70550 Interrupt Register Bit Definitions**

Register	Bit	Bit Definition	Description	Reset Value
IRQ0	7:3	–	<Reserved>	00000
	2	*csma_done_irq*	CSMA done interrupt status	0
	1	*rx_ack_done_irq*	RX acknowledgment process done interrupt status	0
	0	*rx_pkt_done_irq*	RX nonacknowledgment packet process done interrupt status	0
IRQ1	7	*synth_rdy_irq*	Synthesizer ready interrupt status	0
	6	*trim_done_irq*	Trim done interrupt status	0
	5	*trim_fail_irq*	Trimming and tune process failed interrupt status	0
	4	*adc_avg_done_irq*	ADC average done interrupt status	0
	3	*rssi_nosig_irq*	RSSI no signal interrupt status	0
	2	*sync_detect_irq*	Frame synchronization detect interrupt status	0
	1	*pream_det_irq*	Preamble detect interrupt status	0
	0	*rssi_high_irq*	RSSI high interrupt status	0
IRQ2	7	*tx_ack_done_irq*	TX acknowledgment process done	0
	6	*tx_pkt_done_irq*	TX nonacknowledgment packet process done interrupt status	0
	5	*pll_lock_err_irq*	PLL lock error interrupt status	0
	4	*rx_hdr_rdy_irq*	RX header / buffer ready interrupt status	0
	3	*rx_frm_pend_irq*	Frame pending interrupt status	0
	2	*rx_pkt_rdy_irq*	RX packet ready in RX buffer interrupt status	0
	1	*cmd_fail_irq*	SPI MAC command fail interrupt status	0
	0	*cmd_done_irq*	SPI MAC command complete interrupt status	0

5.5 GPIO Pins

There are four general-purpose I/O pins (**GP3..0**). These can be configured as input or output, and as digital or analog. For digital outputs, Table 13, page 32, shows the selection for each signal. The output is 0 for any selector not shown in the table. All 156 digital signals are available at all general-purpose I/O pins using the same code, one code for each general-purpose I/O pin:

- *gpio_0_sel*
- *gpio_1_sel*
- *gpio_2_sel*
- *gpio_3_sel*

To enable a GPIO for output, the corresponding output enable must also be set to 1 in *gpio_oen*. Typically, the input enables should remain disabled with a setting of 0 on each of the *gpio_ien*. In addition, to enable clock output (***_tclk***) signals, *sys_clk_en_test* must be set to 1 in the **CLK_TEST** register.

Table 13 • **GPIO Output Signal Selection**

Selection	Signal	Description
21	cg_sys_tclk	1.2-MHz system clock
22	cg_xtal_tclk	24-MHz crystal oscillator clock
2B	***cg_sys_rst***	System reset
38	*rfmac_synth_tx*	Synth TX: 1 for TX mode;0 for RX mode
3A	trim_fail	Trim fail pulse
3B	trim_done	Trim done pulse
40	dp_tx_data	Transmitter bit data to modulator; includes preamble and frame sync
43	*msc_ptx_en*	PHY transmit operation active
44	dp_pll_lock_err	PLL lock error indication
46	dp_bit_clk_en	Receive and transmit bit clock
47	*ptx_xmtr_en*	TX power amplifier enable status (1 for PA on)
4C	cdr_raw_data	Raw RX data, prior to data recovery and bit alignment
4D	cdr_phy_data	Bit-aligned recovered RX data; includes preamble and frame sync
51	prx_rx_done	Packet receive done status pulse
53	cdr_sync_found	Frame sync pattern detection pulse
54	prx_sync_err	A pulse indicating frame sync timeout or packet abort before frame sync
55	prx_sync_timeout	Frame sync timeout pulse
5A	adc_done	ADC conversion complete
5C	***ptx_tx_trig***	Transmit trigger level signal. Goes high after, programmable time, which starts at the beginning of the transmitted preamble. Time interval is set by *tx_trig_dly_cnt* in 10 us counts. Can be used to enable external PA after synthesizer settling.
63	pream_detected	Indicates preamble detection in RX, and stays high for duration of AFC operation

Table 13 • **GPIO Output Signal Selection *(continued)***

Selection	Signal	Description
69	dp_rx_data	Bit clock aligned RX data to MAC, packet data only, qualified with *dp_rx_active*. Does not include preamble and frame sync.
6A	*dp_rx_active*	RX active signal for payload data; qualifies *dp_rx_data*
6B	*dp_tx_active*	TX active signal for payload data. Goes high after frame sync is transmitted, and stays high while packet data is transmitted from TX buffer.
80	*spis_tx_cmd*	SPI transmit command active status; is high when a SPI TX command sequence is active. Stays high until *msc_cmd_done* status/IRQ is generated.
81	*spis_tx_req_cmd*	SPI data request packet command active status; is high when a SPI TX request command sequence is active. Stays high until *msc_cmd_done* status/IRQ is generated.
82	*spis_rx_cmd*	SPI receive command active status; is high when a SPI RX command sequence is active. Stays high until command sequence is complete or as long the rx_forever is high after SPI receive command is received.
88	*msc_synth_rdy*	Synthesizer ready and PLL locked
89	msc_cmd_done_pulse	A pulse indicating the end of a packet sequence, which generates *cmd_done_irq*
A1	msc_pwrdwn_req	A power-down request from the MAC after command done
A2	*msc_csma_en*	CSMA operation active
A3	*mac_rssi_en*	RSSI block active
A4	*msc_rxc_blk_en*	MAC receive operation active
A5	*msc_txc_blk_en*	MAC transmit operation active
B0	rxc_pkt_rcvd	Packet receive status

Table 13 • **GPIO Output Signal Selection *(continued)***

Selection	Signal	Description
B8	rxc_dup_frm	Duplicate frame received status
B9	*rxc_addr_match*	Address match on received RX frame. Is automatically set if **all** of the following are true: • The destination ID in the received header (*rxc_dest_sid* or *rxc_dest_lid*) matches local device ID (*device_sid* or *device_lid*) • **AND** the source ID in the received header (*rxc_source_sid* or *rxc_source_lid*) matches the unmasked portion of the other ID (*other_sid* or *other_lid*) • **AND** the network ID in the received header (*rxc_network_id)* matches the unmasked portion of the network ID *(network_id*).
BA	*rxc_brdcst_match*	Broadcast address match on received RX frame. This applies to both long and short broadcast mode: • In long addressing mode, the network ID is not used for matching. This bit is set if long addressing mode is used (*rxc_addr_mode* equals 1), **and** broadcast is enabled (*lng_brdcst_en* equals 1), **and** the destination address is all 1s. • In short addressing mode, the unmasked portion of network ID must also match. This bit is set if short addressing mode is used (*rxc_addr_mode* equals 0), **and** broadcast is enabled (*shrt_brdcst_en* equals 1), **and** the destination address is all 1s, **and** the unmasked portion of network ID matches.

6 Setup and Initialization

6.1 Power Management

6.1.1 Power Domains

There are three power domains in the device: VDDD for the digital, VDDA for the analog, and VSUP for the sections of the device that must operate without VDDD and VDDA, including the always-on registers. When the device is in the SLEEP state, VDDD and VDDA are turned off, so all information is lost in those areas at that time.

6.1.2 Power Modes

Table 14, page 35, shows the power states of the ZL70550. Refer to the descriptions in Section 6.2 Power Up, page 35, for the bits that control each state.

To power modes are shown in Table 14, page 35. Stand-by mode requires setting *msc_blk_en* equal to 0. Setting *msc_blk_en* to 1 enables the PLL. The setting of *msc_blk_en* during sleep determines whether or not the PLL is turned on wake-up.

Table 14 • **Power Modes and Operational Conditions**

State	VDDD	XTAL Osc	*cg_sys_clk*	PLL	TX Circuit	RX Circuit	Current
SLEEP							10 nA
IDLE[1]	✓	✓	✓				150 µA
STANDBY[1]	✓	✓	✓	✓			700 µA
ACTIVE TX	✓	✓	✓	✓	✓		2.75 mA
ACTIVE RX	✓	✓	✓	✓		✓	2.4 mA

[1] *msc_blk_en* affects whether the device goes into IDLE or STANDBY.

6.2 Power Up

6.2.1 Initial Reset

When VSUP power is applied to the device, **RESET_B** should be asserted low to initialize the device into its default state. After **RESET_B**, the device is in the SLEEP state, in which the VDDD section of the device is powered-down. VDDD is the regulated voltage domain that is supplied by the VSUP voltage domain, which is the main external power to the device. **RESET_B** also resets all of the registers in the device.

6.2.2 Wake-Up Operation

The only mechanism for waking up the device is to assert or pulse the **SPI_SEL_B** line low. This starts the wake-up sequencer, which powers up the VDDD section. When the VDDD power-up sequence is complete, the **IRQ** line goes high (if enabled) and the **SPI_MISO** line goes high if **SPI_SEL_B** is low. This indicates that the SPI interface is ready to accept commands and read/write operations. The IRQ is cleared on the first rising edge of **SPI_CLK**, so no explicit clearing of this interrupt is required. While the device is not ready, **SPI_MISO** remains low while **SPI_SEL_B** is asserted low before the first **SPI_CLK**.

6.2.3 Monitoring Wake-Up

To monitor wake-up without asserting the **SPI_SEL_B** line low, ***cg_spi_rst*** can be monitored from one of the GPIO pins. The signal ***cg_spi_rst*** is high if VDDA and VDDD are powered up, and the device is ready to receive SPI operations, and the **SPI_SEL_B** pin is high. The signal ***cg_spi_rst*** is selected as the GPIO output by writing 0x2D to one of the GPIO selector registers along with the corresponding output enables. These registers remain unchanged during a SLEEP event, and the selected output is activated as soon as VDDD power is stable and the internal ***cg_sys_rst*** is released.

If ***cg_spi_rst*** is selected as an output on GPIO, then the level of that GPIO indicates that whether VDDD and VDDA are fully powered-up, the crystal oscillator is running, and the digital reset has been released.

Note: The *trim_done_irq* status is set after wake-up. This status should be cleared and ignored.

6.3 Sleep Operation

Powering down VDDD causes the device to go into the SLEEP state. The VDDD section is normally powered down manually but can be powered down automatically if desired.

6.3.1 Manual Sleep Operation

The VDDD section is normally powered-down by writing the power-down command to the SPI interface. This is done by writing 0xDE to **SPIR_PWRDWN_REQ** using a short-address write command.

6.3.2 Automatic Sleep Operation

Two options are available for automatically putting the device into the SLEEP state if desired.

- If *mac_pwrdwn_en* equals 1 and no interrupt is currently active, an SPI command done status from the MAC can cause the device into the SLEEP state.
- If *dp_pwrdwn_en* equals 1, then a low RSSI level after the receiver is enabled causes the device to SLEEP if no interrupt is currently active.

If the interrupt is asserted at the time of the SLEEP event, then the SLEEP request is blocked. However, care must be taken in responding to an interrupt when automatic SLEEP is enabled.

The SLEEP event from the MAC only occurs at *cmd_done_irq*, but *cmd_done_irq_en* is 0 for automatic SLEEP so no interrupt is generated on that condition. If another interrupt occurs prior to *cmd_done_irq*, such as *rx_pkt_rdy_irq* or *rx_frm_pend_irq*, these interrupts can occur before *cmd_done_irq*. If the interrupt status is read while the automatic SLEEP is enabled, then the sleep event may occur while the interrupt is being processed. Therefore, it is recommended to disable automatic sleep by setting *mac_pwrdwn_en* and *dp_pwrdwn_en* equal to 0 prior to reading interrupt status.

6.4 Synthesizer Controller and Channel Selection

The A and M registers must be programmed any time a different channel is desired.

The A value can be programmed through the system bus using *a_div* in the **SYNTH_A_DIV** register.

The M value can be programmed through the system bus using **SYNTH_M_DIV**.

The M value and the A value should be written as a two-byte write operation starting at the address for the **SYNTH_M_DIV** register.

LO control (high/low) can be programmed through the system bus using *ch_lo_ctrl* in the **VCO_CTRL** register. This bit controls whether the mixing IF is above or below the channel frequency, and determines where the channel image is located during receiver operations.

6.4.1 A and M Requirements

The A and M requirements are:

$$A \geq 5,\ M \geq 16 \qquad \textit{EQ 6-3}$$

6.4.2 A and M Value Calculation

The total number of PLL clock periods to be produced by the VCO in order to run through the PLL divide cycle is given by N_{tot}:

$$N_{tot} = \text{int(carrier frequency / PLL clock)} \qquad \textit{EQ 6-4}$$

In the ZL70550 PLL implementation, the relation between N_{tot} and A and M is given by EQ 6-5 below:

$$N_{tot} = 17 \times A + 16 \times M \quad \textit{EQ 6-5}$$

EQ 6-5 shows that A is the number of times the prescaler needs to count to 17 and M is the number of times the prescaler needs to count to 16.

$$A = ((N_{tot} - 5) \bmod 16) + 5 \quad \textit{EQ 6-6}$$

where: $4 < A < 21$

$$M = (N_{tot} - A \times 17) / 16 \quad \textit{EQ 6-7}$$

where: $139 < M < 193$ for 300-kHz channel spacing

EQ 6-4, EQ 6-6, and EQ 6-7 can be used for calculating A and M values to program registers **SYNTH_A_DIV** and **SYNTH_M_DIV**.

6.4.3 Channel Frequency Calculation

The *ZL70550 Synthesizer Programming Table.xls* spreadsheet implements the above formulas and can also be used to generate the A and M values for a particular target frequency or as a look-up table for all the synthesizable frequencies within the ZL70550 range.

The channel frequency is:

$$((A \times 17) + (M \times 16)) \times 300\ \text{kHz} \quad \textit{EQ 6-8}$$

where: 300 kHz is the channel width

6.4.4 A and M Programming Example

1. Determine the output frequency for the current channel being programmed. N_{tot} represents the number of counts that corresponds to the output frequency. From the table, look up the values for A and M. A is the number of times the prescaler divides by 17 and M is the number of times the prescaler divides by 16.
2. If a frequency of 915.9 MHz is desired, this would equate to an N_{tot} value of 3053.
3. If N_{tot} of 3053 is chosen, the values in EQ 6-8 , page 37, are 17 × 13 + 16 × 177 = 3053, that is:
   - A is 13, with a binary representation of 6’b001101.
   - M is 177, with a binary representation of 8’b10110001.
4. For this example, *vco_low_range* remains at 0. If channels below 795 MHz are being used, then program the *vco_low_range* bit in the **VCO_CTRL** register to 1. Programming a 1 shifts the VCO trim range so that the PLL can take the VCO down to 779 MHz.
5. Program *m_div* equal to 8’b10110001 (0xB1) in the **SYNTH_M_DIV** register.
6. Program *a_div* in **SYNTH_A_DIV** equal to 8’b00001011 (0x0B).

6.5 Recommended Initialization Settings

Chapter 8 Registers, page 52, summarizes the memory map of the ZL70550. Recommended initial register settings are provided in the "Recommended Values" column of Table 35 in Section 8.2 Address Space, page 53. Write the recommended values to the appropriate registers after every chip reset (**RESET_B** pin low).

Before packet transactions can be performed, the device must first be trimmed for the channel of operation, and then the packet parameters must be set. Basic variables needed are number of FCS (CRC) bytes, length of preamble, DC restore value and mode, and preamble detect mode.

6.5.1 Z-Star Operation

For Z-Star operation, the following values must be set. Any value that is not shown here or in Section 8.2 Address Space, page 53, should use the default (reset) value.

Table 15 • **General Parameters**

Parameter	Node	Hub	Default	Function
msc_blk_en	1	1	0	MAC enable and turn on the PLL
hub_node_n	0	1	0	Hub enable: 1 for hub, 0 for node
tx_fsn_incr_dis	0	1	0	Disable auto TX frame sequence number increment

Table 16 • **Receive Parameters**

Parameter	Node	Hub	Default	Function
lng_brdcst_en	0	1	0	Enable receiving Z-Star long-address broadcast
dup_frm_irq_en	0	1	0	Enable packet ready interrupt on duplicate frame
rx_forever	0	1	0	Receive continuously
addr_mask	0xFFFF	0xFF00	0xFFFF	Mask for received Z-Star source & network ID

6.5.2 For Raw Bit Mode

6.5.2.1 TX Raw Bit Mode

Sources for TX data are either the TX buffer or pin **GP1**. For test purposes, the frame sync pattern may also be continuously transmitted. In raw bit mode, the transmitter transmits only raw data, without preamble and frame sync. If preamble and frame sync are needed, then they must be supplied by the raw data source.

When *gpio_tx_sel* equals 1, the bit rate is controlled by the data source at **GP1**. To use the device bit clock, the signal ***dp_bit_clk_en*** may be output to one of the other **GP3..0** pins using selection code 0x46 as described in Section 5.5 GPIO Pins, page 32.

Table 17 • **TX Raw Bit Mode Controls**

		Recommended	
Field	**Memory Map Location**	**Value**	**Comments**
tx_mode	**TX_CTRL0[2:0]**	000	Selects raw bit mode TX format
raw_tx_mode	**PHY_TX_RAW_MODE_CTRL[1]**	1	Enables TX operation in raw bit mode
tx_buf_len	**TX_BUF_LEN0..1**	Length	Sets byte length of TX
dp_sync_always	**DP_CTRL0[2]**	0 or 1	Optional. Set to 1 for continuous TX of sync pattern
gpio_tx_sel	**PHY_TX_RAW_MODE_CTRL[0]**	0 or 1	Optional; set to 1 to select **GP1** as TX data source
gpio_ien	**PAD_EN0[3:0]**	0 or 0x2	Optional; set to 0x2 if using **GP1** as TX data source. Enables **GP1** input.

Table 17 • **TX Raw Bit Mode Controls**

Field	**Memory Map Location**	**Recommended Value**	**Comments**
gpio_oen[1]	**PAD_EN0[5]**	0	Optional; always set to 0 if using **GP1** as TX data source. Disables use of **GP1** for output.
gpio_X_sel	**GPIO_X_SEL** (where X is 3, 2, or 0, corresponding to unused pin **GP3, GP2**, or **GP0**, respectively)	0 or 0x46	Optional; if using pin **GP1** as TX data source, writing 0x46 assigns any unused **GP3..0** pin as output for device bit clock (***dp_bit_clk_en***)
gpio_oen	**PAD_EN0[7], PAD_EN[6],** or **PAD_EN0[4]** (corresponding to unused pin **GP3, GP2**, or **GP0**, respectively)	Any	Optional. Enable any unused **GP3..0** pin if using **GP1** as TX data source AND outputting device bit clock (***dp_bit_clk_en***) to GPIO. Enable the pin corresponding to the GPIO selection register (*gpio_X_sel*) used for outputs above.

6.5.2.2 RX Raw Bit Mode

In RX raw bit mode, all data bits received are output to the RX buffer in non-bit-aligned bytes. Preamble detection is recommended, but not required (if not used, then *pream_det_mode* is set to 0). No frame sync is required.

The raw bit data may also be received on **GP1**, either as ***cdr_raw_data*** using GPIO selection code 0x4C, or as ***dp_rx_data*** using GPIO selection code 0x69 as described in Section 5.5 GPIO Pins, page 32. The signal ***dp_rx_data*** uses full bit recovery and is aligned to ***dp_bit_clk_en*** (which is used on both TX and RX; see above).The signal ***cdr_raw_data*** is raw receiver data and not bit aligned.

Table 18 • **RX Raw Bit Mode Controls**

Field	**Memory Map Location**	**Recommended Value**	**Comments**
rx_mode	**RX_CTRL0[2:0]**	000	Selects raw bit mode for receiving packets
raw_rx_mode	**PHY_RX_MODE_SEL[3]**	1	Enables RX operation in raw bit mode
rx_frm_len	**RX_FRM_LEN0/1**	Length	Sets byte length of RX
pream_det_mode	**PHY_RX_MODE_SEL[0]**	0 or 1	Recommended that this bit be set to 1 to enable preamble packet detection
gpio_X_sel	**GPIO_X_SEL** (where X is 3, 2, or 0, corresponding to unused pin **GP3, GP2**, or **GP0**, respectively)	0 or 0x4C or 0x69	Optional. Write 0x4C or 0x69 to assign an unused GPIO pin for RX data as ***cdr_raw_data*** or ***dp_rx_data***, respectively.
gpio_X_sel	**GPIO_X_SEL** (where X is 3, 2, or 0, corresponding to unused pin **GP3, GP2**, or **GP0**, respectively)	0 or 0x6A	Optional. Write 0x6A to select any unused **GP3..0** pin to qualify ***dp_rx_data*** if desired.
gpio_oen	**PAD_EN0[7:4]**	Any	Optional. Enable appropriate **GP3..0** pin(s) if using GPIO as RX output. Enable the pin(s) corresponding to the GPIO selection register(s) (*gpio_X_sel*) used for outputs above.

7 Calibrations

7.1 Overview/Summary

Calibration requirements are needed for a variety of subsystems in the ZL70550 to optimize parameters controlling the frequency of oscillators, antenna matching, filters, etc.

The different calibrations and optimizations in the ZL70550 are described in Section 7.2 Sequence, page 41, and Section 7.5 Procedures, page 43. The majority of calibrations require no external equipment and may be performed very quickly by the ZL70550 itself (refer to Section 7.4 Required Time for Calibrations, page 43). Section 7.5 Procedures, page 43, specifies the unique requirements of each calibration.

It is optional (with an always-on setup bit) to automatically tune the RC oscillator after power-on reset. Otherwise the internal trims are executed only on command.

ZL70550 does not perform any trimming or tuning at wake-up or at any other time without being commanded to do so over the control interface.

Any combination of automated calibrations can be initiated by the user application writing commands via the SPI bus interface. No further user intervention is then needed until the ZL70550 signals to the user application that all requested calibrations are complete.

Registers (or parameters) that control analog circuit operation may be categorized into three classes:

- Preset: Registers whose values are defined during IC evaluation and are set to a constant value (ideally the default value). No calibration is required for such parameters. These registers are supplied for design flexibility and to reduce risk.
- Factory: Registers whose values must be determined in production calibration procedures requiring special equipment. These registers relate to parameters that may vary from device to device due to process variations, operation range selections, or external component values.
- Operation: Registers whose values must be determined in production and/or by operational calibration procedures executed by the chip independent of external equipment. These registers relate to parameters that may vary from device to device due to process variations, operation range selections, or external component values.

Parameters are typically stored in internal always-on memory. Because the internal always-on memory depends on application of a battery voltage on the **VSUP** pin, SLEEP and wake-up of the device does not affect the always-on memory. However, if the chip loses battery voltage or if the device is reset using the **RESET_B** pin, it is necessary to retrim or reload the parameters from external storage.

7.2 Sequence

The calibration sequence should be performed on every power-up on untrimmed, uncalibrated parts in the order provided in Table 19, page 41. Only one of the calibrations at a time can be performed by the application software. Calibrations marked with “F” should be done at the factory and then stored in external nonvolatile memory. These values should be loaded to the device on subsequent power-ups.

Table 19 • **Recommended System Calibration Order**

Order	Description	P (Preset) F (Factory) O (Operating)	Supported by TRIM_CMD	Notes
1	Current reference	O	X	
2	Crystal oscillator frequency	F		
3	RC oscillator	O	X	1
4	VCO amplitude	O	X	1,2
5	VCO full frequency	O	X	1,2
6	Peak detector offset	O	X	1,2
7	Antenna	O	X	3
8	LNA load	O	X	3
9	RX filter	O	X	
10	FSK deviation	O	X	1
11	Transmitter output power	P,F		4
12	LNA gain	P,F		4

1. Periodic retrim may be desired due to thermal change.
2. It is recommended that these calibrations be performed anytime there is a frequency change. The *vco_low_range* bit must be set by application prior to tune if frequency is less than 815MHz. Retrim is also prescribed if a *trim_fail_irq* or *pll_lock_err_irq* is triggered.
3. Whenever the frequency changes, it may be necessary to run these calibrations to maintain optimum performance. It is recommended that this be determined during product characterization. These tests require that the TX and RX pins be connected together. Note that the antenna tune capacitors are only on the RX pins and that the LNA load tune requires a transmitted signal fed back to the receiver.
4. Preset trim values are determined during product characterization and should be the same for all devices

The calibration sequence should be executed in the order provided in Table 19, page 41 and in the following paragraphs. At the end of most tune or trim procedures, the *trim_done_irq* interrupt is set. The **IRQ1** register should then be read to see if the *trim_fail_irq* status bit is set. The **IRQ1** register is cleared on the read operation.

Note: **IMPORTANT**: In order to optimize the tune and trim procedure on the ZL70550 chip at the factory or after every power-up, the tune and trim sequence described in the following paragraphs assumes that the steps are performed in a contiguous manner. If a tune or trim has to be run independently, the settings performed before this particular tune or trim have to be taken into account.

7.3 Setup, Initiation, and Control of Calibrations

The tune and trim sequence can start only after setting up the basic functionality of the ZL70550 chip both at the factory and at power-up. Refer to Table 20, page 42.

Table 20 • **Procedure for Tune and Trim Setup**

1. Power up and wake-up device in accordance with Section 6.2 Power Up, page 35.	
2. Write **IRQ_EN1[6]** = 1.	This enables the *trim_done_irq* interrupt.

A write to the control register **TRIM_CMD** starts an automatic trim algorithm if any previous command was completed. If the previous command is not done, any writes are ignored and trimming not executed. Table 21, page 42, defines the command value to be written for each of the available automatic trims. Executing a trim command unconditionally and asynchronously disables any on-going or new MAC operations until the trim command is done. Application software is responsible for assuring orderly transitions between normal media operation and trimming. It is advisable that any external processes that can generate SPI writes to the chip be temporarily disabled until a trim is completed to avoid interference with a trim.

Termination of a trim command is indicated by the *trim_done_irq* status. This can be polled or an interrupt generated when the *trim_done_irq_en* bit is set. If a trim failure was encountered by the trim command, it is indicated on the *trim_fail_irq* status bit when *trim_done_irq* is read.

There are a handful of setup bits that may be useful for certain applications and trims. See trim control registers in memory map. This need to be written before initiating the relevant trim. Application of some setup bits may depend on lab and manufacturing evaluation results.

Table 21 • **Trim Commands**

Command	Algorithm	Typical Customer Prerequisites
0	No-op	
1	Current reference	None
2	VCO full frequency	Channel frequency
6	VCO amplitude	Channel frequency
7	FSK frequency separation	Valid full VCO frequency trim
8	Blocker peak detector offset	None
9	LNA load	Peak detector trim, full VCO frequency trim
10	Antenna	LNA load tune
11	RX filter (IF filter, FM detector, and Gaussian filter)	VCO full frequency trim
12	RC oscillator	None

7.4 Required Time for Calibrations

The length of time for each internal automated calibration is defined in Table 22, page 43.

Table 22 • **Required Time for Calibrations**

Calibration	Typical Time
Current reference	400 µs
VCO full frequency	150 ms[1]
VCO amplitude	500 µs
FSK frequency separation	500 ms
Blocker peak detector offset	3 ms
LNA load	12 ms
Antenna	12 ms
RX filter (IF filter, FM detector and Gaussian filter)	3 ms
RC oscillator	1 ms

[1] Time depends on whether correct frequency band was selected prior to initiating the VCO full frequency trim.

7.5 Procedures

7.5.1 Current Reference Trimming

It is very important to trim the current reference before any other trimming or tuning because the current reference supplies current to almost all the analog circuits on the chip. If the reference is untrimmed then the other trims are not valid.

The ZL70550 uses an external resistor of 2% or better accuracy to set its internal current reference. The node attached to the external resistor is sensitive to noise that can be picked up by the exposed pin and trace runs. For this reason, internal resistors are used in the current reference and other reference circuits. Before they can be used, however, an internal resistor of the same type and value is compared with the external resistor and trimmed to match.

The current reference trimming is described in Table 23, page 43. It assumes that the procedure in Section 7.3 Setup, Initiation, and Control of Calibrations, page 42, has already been completed.

Table 23 • **Procedure for Current Reference Trim**

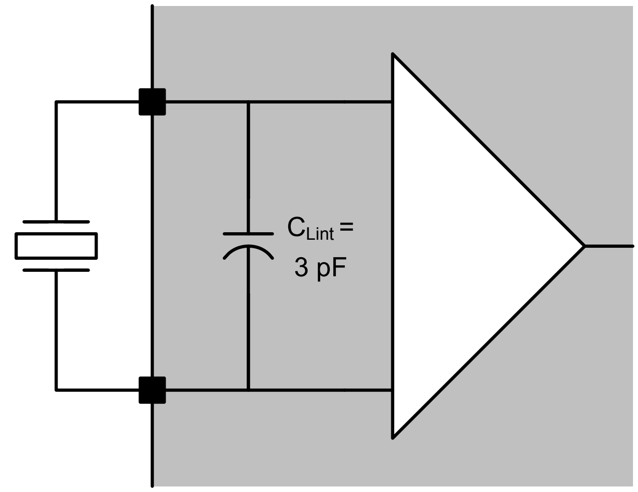
1. Perform the following setup through the control interface:	
a. Write **TRIM_CMD** = 0x01.	
2. Now that the automatic IREF trimming procedure is launched:	
a. Wait for the *trim_done_irq*.	Indicates that the trim has completed. The trim typically takes less than 500 µs.
3. At the end of the IREF trimming procedure:	
a. Read **IRQ1**.	Read the **IRQ1** register to clear interrupt(s) and ensure that the trim did not fail.

7.5.2 XO Tuning

The purpose of the crystal oscillator tuning is to improve the absolute accuracy of the system reference frequency. This tuning is done once during factory calibration.

Note: **IMPORTANT**: The crystal oscillator tuning depends on the selection of the crystal and the loading that is placed on the crystal pins. In order to save power, the crystal oscillator presents a 3-pF load instead of the typical 8-pF or 10-pF load. A slight frequency pull, on the order of 100 to 150 PPM, would result if using a standard crystal without additional external load capacitors. Such a deviation has no effect on the operation of the device and is generally not a problem for most applications, providing all ZL70550 devices have the same frequency pull (within trimmable range). If the deviation is not acceptable and power is critical, a special cut crystal may be used (that is, slightly slower to compensate for the pull). Microsemi is engaging with crystal manufacturers in developing custom crystals that operate at 24 MHz with only a 3-pF load.

Figure 21 • **Crystal Oscillator with Optional Additional External Load Capacitors**



0026~Xtal diagram~v1606.0

The crystal oscillator tuning procedure is described in Table 24, page 44.

Table 24 • **Procedure for Crystal Oscillator Tune**

1. Route out the crystal frequency to desired GPIO pin (**GPIO3..0**): a. Set the corresponding output enable bit in the **PAD_EN0** register. b. Write 0x22 to the corresponding GP input/output register (**GPIO_3..0_SEL**). c. Write 0x08 to the **CLK_TEST** register.	
2. Measure actual crystal (F_{meas}).	Use a frequency counter that has better than 1 PPM accuracy.
3. Compare desired frequency (F_{des}) to measured frequency (F_{meas}). a. If $F_{meas} < F_{des}$, then write **XO_TRIM =** − 1. b. Else: If $F_{meas} > F_{des}$, then write **XO_TRIM = XO_TRIM** + 1. c. Else: If $F_{meas} = F_{des}$, then trim is complete.	Change the six-bit control word in the **XO_TRIM** register until the desired crystal frequency (F_{des}) is reached. Desired frequency should be as close to 24 MHz as possible.

Table 24 • **Procedure for Crystal Oscillator Tune *(continued)***

4. Check whether **XO_TRIM** is out of range. a. If **XO_TRIM** = 0x00 or if **XO_TRIM** = 0x3F, then the trim has failed. b. Else: Repeat steps 2 and 3.	Repeat steps 2 and 3 until desired crystal frequency is reached or trim fails.
5. Store **XO_TRIM** value in external nonvolatile memory.	Since the crystal oscillator trim is a factory trim, the trim value has to be stored in external nonvolatile memory and loaded into the **XO_TRIM** register upon power-up of the device.

7.5.3 RCO Tuning

The purpose of the 150-kHz strobe oscillator tuning is to set the output frequency to 150 kHz for accurate timing of all processes controlled by this oscillator.

Table 25 • **Procedure for RC Oscillator Trim**

1. Perform the following setup through the control interface:	
a. Write **TRIM_CMD** = 0x0C.	
2. Now that the automatic RC oscillator trimming procedure is launched:	
a. Wait for the *trim_done_irq*.	Indicates that the trim has completed. The trim typically takes less than 500 µs.
3. At the end of the RC oscillator trimming procedure:	
a. Read **IRQ1**.	Read the **IRQ1** register to clear interrupt(s) and ensure that the trim did not fail.

7.5.4 VCO Frequency and Amplitude Trimming

The terms "tuning" and "trimming" are used interchangeably in this section. Please read all the sections related to VCO trimming before starting programming.

Trimming the VCO frequency and amplitude keeps the VCO in the proper operating conditions. There are several steps to accomplish this and they should be done in the order shown:

1. Select the frequency band (see Section 7.5.4.1 Frequency Band Selection, page 45).
2. Perform VCO full frequency tuning and VCO amplitude trimming (see Sections 7.5.4.2 VCO Amplitude Trimming, page 46, and 7.5.4.3 VCO Frequency Trimming, page 47):

Detailed descriptions of the trim steps follow this section.

Note: **IMPORTANT**: Periodic retrimming of the VCO may be required if the antenna impedance changes or operating conditions change drastically.

Note: **IMPORTANT**: Trim the VCO with the expected antenna in the nominal application condition.

The VCO frequency can be trimmed for every channel and the resulting values stored in a table. When a channel is selected, the appropriate trim values can be written back to the VCO. We recommend doing all of the calibrations upon any change of frequency.

7.5.4.1 Frequency Band Selection

Before any automatic frequency tuning is initiated, the frequency range must be selected. An initial, rough range setting for the *vco_low_range*, and *vco_frq_band_trim* bits can be found in Table 26, page 46. IC process variations affect these ranges. The first step is to pick the correct frequency range.

There are two ranges, a low range and a high range. If you are using the device between 815 MHz and 965 MHz, set the *vco_low_range* bit to 0. If you are using the device between 779 MHz and 815 MHz, set the *vco_low_range* bit to 1.

The second step is to pick the frequency band. The *vco_frq_band_trim* value is used to select the frequency band. The first and second columns in Table 26, page 46, show the band settings. At the tops of these columns are the register names and bit positions in the registers. The frequencies vary with temperature, voltage and process so they are offered as guidance, not guaranteed. Select the band in which your frequency of operation is nearest to the center of the band or a little higher than center.

Table 26 • **Typical Frequency Range Selection**

vco_low_range **(at VCO_CTRL[5])**	***vco_frq_band_trim***	**Frequency (MHz)**
1	1	768.0 – 788.0
1	2	779.2 – 800.0
1	3	790.4 – 812.4
1	4	802.8 – 825.6
0	0	813.2 – 836.8
0	1	826.0 – 850.8
0	2	837.6 – 863.2
0	3	851.6 – 878.8
0	4	862.0 – 890.4
0	5	877.6 – 907.2
0	6	890.8 – 921.6
0	8	903.6 – 935.6
0	7	907.6 – 940.4
0	9	921.2 – 955.6
0	10	934.4 – 970.0
0	11	954.4 – 992.4

7.5.4.2 VCO Amplitude Trimming

The VCO amplitude trimming procedure is described in the table below.

Table 27 • **Procedure for VCO Amplitude Trim**

1. Perform the following setup through the control interface:	
a. Write **SYNTH_M_DIV** and **SYNTH_A_DIV** per instructions in Section 6.4 Synthesizer Controller and Channel Selection, page 36.	Set the synthesizer A and M values to the operational frequency bank (for more details, refer to Section 6.4 Synthesizer Controller and Channel Selection, page 36).
b. If necessary, set *vco_low_range* in **VCO_CTRL** per Table 26, page 46.	Writing to the *vco_low_range* selects the frequency range. (See Table 26, page 46.)
c. Write **VCO_FRQ_BAND_TRIM** per Table 26, page 46	Writing to *vco_frq_band_trim* selects the frequency band. (See Table 26, page 46.)
2. Write **MAC_CTRL[1]** = 0.	Clearing the *msc_blk_en* bit performs a synchronous reset of the MAC.

Table 27 • **Procedure for VCO Amplitude Trim *(continued)***

3. Write **TRIM_CMD** = 0x06.	Writing 0x06 to *trim_cmd* launches the automatic VCO amplitude trim. The **TRIM_CMD** register is cleared internally once the trim procedure is done.
4. Now that the VCO amplitude trimming procedure is launched:	
a. Wait for *trim_done_irq*.	Indicates that the trim has completed. This trim takes approximately 500 us.
5. At the end of the VCO amplitude trimming procedure:	
a. Read **IRQ1**.	Read to clear the **IRQ1** register and to see whether the *trim_fail_irq* status bit has been set.

7.5.4.3 VCO Frequency Trimming

There are four always-on registers used to set *vco_frq_trim* for a currently selected channel. The *vco_frq_band_trim* field sets the upper four bits for “band” frequency trimming. Either the *vco_frq_rx*, *vco_frq_txpaoff*, or *vco_frq_txpaon* field sets the lower eleven bits for “fine” frequency trimming. The latter three fields account for the intermediate synthesizer frequency offset necessary for reception and for possible PA load differences on the VCO output depending on whether or not the PA is turned on. The selection multiplexer is integrated into the ZL70550 and is automatically controlled by the state of the radio.

A VCO full frequency trim command executes all three fine frequency trim commands while also finding a trim band. The ZL70550 register fields affected by the VCO full frequency trim are *vco_frq_rx*, *vco_frq_txpaoff*, *vco_frq_txpaon*, and *vco_frq_band_trim.*

7.5.4.3.1 VCO Full Frequency Trimming

The VCO full frequency trimming procedure is described in the table below

Table 28 • **Procedure VCO Full Frequency Trim**

1. Perform the following setup through the control interface:	
a. Write **TRIM_CMD** = 0x02.	Writing 0x02 to *trim_cmd* launches the full VCO frequency tune. The **TRIM_CMD** register is cleared internally once the trim procedure is done.
2. Now that the VCO full frequency trimming procedure is launched:	
a. Wait for the *trim_done_irq*.	Indicates that the trim has completed. If the *vco_frq_band_trim* is not set to the correct band before the trim is launched, the internal algorithm will attempt to find the correct frequency band and re-trim the VCO. Each iteration of the full VCO frequency trim takes approximately 150 ms.
3. At the end of the VCO full frequency trimming procedure:	
a. Read **IRQ1**.	Read the **IRQ1** register to clear interrupt(s) and ensure that the trim did not fail.

7.5.5 Blocker Peak Detector Offset Trimming

DC offset is created in the blocker peak detector by a process dependent resistance. The offset is controlled by a five-bit trim DAC. With no signal present at the input DC offset can be trimmed to 0 at the output. Trimming utilizes the five-bit ADC connected to the peak detector output. The trimming procedure for the peak detector offset is described in Table 29, page 48.

Table 29 • **Procedure Blocker Peak Detector Offset Trim**

1. Perform the following setup through the control interface:	
a. Write **TRIM_CMD** = 0x08.	Writing 0x08 to *trim_cmd* launches the blocker peak detector offset trim. The **TRIM_CMD** register is cleared internally once the trim procedure is done.
2. Now that the blocker peak detector offset trimming procedure is launched:	
a. Wait for the *trim_done_irq*.	Indicates that the trim has completed. The trim typically takes less than 3 ms.
3. At the end of the blocker peak detector offset trimming procedure:	
a. Read **IRQ1**.	Read the **IRQ1** register to clear interrupt(s) and ensure that the trim did not fail.

7.5.6 Antenna Tuning

The purpose of antenna tuning is to select the capacitance, which peaks the antenna resonance at the required frequency. A peak detector in the RF receiver section stores the maximum voltage swing on the receiver input pins. The voltage is measured by the five-bit ADC block. An algorithm for finding the best capacitor value is implemented in the trim algorithms.

Antenna tuning is performed during manufacturing after the device is mated with an antenna, after any channel frequency bank switch, and after any power-up if the trim value is not stored. It is best to store the known good trim value and restore it to the trim register at power-up. The antenna tuning procedure is described in Table 30, page 48.

Table 30 • **Procedure for Antenna Tuning**

1. Write **LNA_GAIN** = 0x00.	Sets the lowest internal LNA gain.
2. Write **LNA_BIAS_TRIM** = 0x01	Set the *lna_bias_trim* bits to 4’b0000 to set the LNA bias trim to its lowest setting.
3. Write **PA_PWR_LEVEL** = 0x80	Sets the power amplifier to the lowest power setting by writing 6’b00 0000 to *pa_pwr_level[5:0]*.
4. Write **TRIM_CMD** = 0x0A.	Writing 0x0A to *trim_cmd* launches the antenna trim. The **TRIM_CMD** register is cleared internally once the trim procedure is done.

1. If the *trim_fail_irq* status has been set by the tuning procedure, read the trim value in *ant_trim*:

   If the tune frequency is at the high end of what the VCO can reach and the trim value is at the lowest trim value, then there is no failure.

   If the tune frequency is at the low end of what the VCO can reach and the trim value is at the highest trim value, then there is no failure.

   Check the value of the ADC during the trimming (in the **ANT_PEAK_ADC** register). If the value is below 5 or above 26 then adjust the transmitter output power accordingly. If the value in *ant_peak_adc* is below 5, increase the transmitter output power by 1; if the value is over 26, decrease the PA power level by setting *pa_pwr_level[6]*. Then rerun this tune.

Table 30 • Procedure for Antenna Tuning *(continued)*

5. Now that the antenna tune procedure is launched:	
a. Wait for *trim_done_irq*.	Indicates that the trim has completed. The trim typically takes less than 15 ms.
6. At the end of the antenna tune procedure:	
a. Read **IRQ1**.	Read to clear the **IRQ1** register and to see whether the *trim_fail_irq* status bit has been set.
b. If **IRQ1[5]** = 1, then the trim may have failed; see Note 1.	The *trim_fail_irq* status has been set by the tuning procedure when the **ANT_TRIM** register is set to 0 (minimum) or 31 (maximum). See Note 1.

1. If the *trim_fail_irq* status has been set by the tuning procedure, read the trim value in *ant_trim*:

   If the tune frequency is at the high end of what the VCO can reach and the trim value is at the lowest trim value, then there is no failure.

   If the tune frequency is at the low end of what the VCO can reach and the trim value is at the highest trim value, then there is no failure.

   Check the value of the ADC during the trimming (in the **ANT_PEAK_ADC** register). If the value is below 5 or above 26 then adjust the transmitter output power accordingly. If the value in *ant_peak_adc* is below 5, increase the transmitter output power by 1; if the value is over 26, decrease the PA power level by setting *pa_pwr_level[6]*. Then rerun this tune.

7.5.7 LNA Load Tuning

To get enough gain in the front-end amplifier without using much current, an inductive load is tuned to resonate at the receive frequency. The load tuning capacitance must change when the frequency bank changes, and the LNA load tune section of the tune and trim block is required to find the optimum tuning capacitance for a given channel. The tune block saves the setting with the largest peak detector response and restores that setting at the end of the routine.

7.5.7.1 Execute LNA Load Tuning

The LNA load tuning procedure is described in Table 31, page 49. The following steps assume the setup for the antenna tune was done prior to running this trim.

Table 31 • Procedure for LNA Load Tune

1. Write **TRIM_CMD** = 0x09.	Writing 0x09 to *trim_cmd* launches the LNA load trim. The **TRIM_CMD** register is cleared internally once the trim procedure is done.
2. Now that the LNA load tune procedure is launched:	
a. Wait for *trim_done_irq*.	Indicates that the trim has completed. The trim typically takes less than 15 ms.
3. At the end of the LNA load tuning procedure:	
a. Read **IRQ1**.	Read to clear the **IRQ1** register and to see whether the *trim_fail_irq* status bit has been set.
b. If **IRQ1[5]** = 1, then the trim *may* have failed; read *lna_frq_trim* and see Note 1.	If the *trim_fail_irq* status has been set by the tuning procedure, read the trim value in *lna_frq_trim* and see Note 1.

1. If the *trim_fail_irq* status has been set by the tuning procedure, read the trim value in *lna_frq_trim*:

   If the selected frequency range is at the high end of what the VCO can reach and the trim value is at the lowest trim value, then there is no failure. The trim value is appropriate.

   If the tune frequency selected is at the low end of what the VCO can reach and the trim value is at the highest trim value, then there is no failure. The trim value is appropriate.

   Check the value of the ADC during the tuning (in the **LNA_PEAK_ADC** register. If the value is below 5 or above 26 then adjust the transmitter output power. If the value in *lna_peak_adc* is below 5, increase the transmitter output power by 1; if the value is over 26, decrease the PA power by 1. Then rerun this tune.

7.5.8 RX Filter (FM detector, IF filter, and Gaussian filter) Trimming

The IF filter is critical to rejection of adjacent channel interference and must be tuned to achieve acceptable performance. The Gaussian filter characteristics are critical for low-side lobes in the transmitter output spectrum. The IF filter, the FM detector, and the Gaussian filter are designed with a similar topology, with component trimming scaled so that they can be tuned in parallel. The tune register values are at the same time used for the IF filter and Gaussian filter, ensuring that the IF filter's center frequency is at the IF and the Gaussian filter characteristics are as designed. This trim also handles the process, temperature, and voltage variation for the FM detector by determining and then writing an optimal value to the **DC_CNTR_TRIM** register. This trim value provides the starting point for the FM detector adjustments that are made automatically during the preamble of an incoming data packet.

The RX filter tuning procedure is described in Table 32, page 50.

Table 32 • **Procedure for RX Filter Tune**

1. Write **TRIM_CMD** = 0x0B.	Writing 0x0B to *trim_cmd* launches the RX filter trim. The **TRIM_CMD** register is cleared internally once the trim procedure is done.
2. Now that the RX filter trim procedure is launched: a. Wait for *trim_done_irq*.	 Indicates that the trim has completed. The trim typically takes less than 15 ms.
3. At the end of the RX filter trim procedure: a. Read **IRQ1**.	 Read to clear the **IRQ1** register and to see whether the *trim_fail_irq* status bit has been set.

7.5.9 FSK Deviation Trimming

The gain of the direct modulating signal must correspond to the frequency bank being used. The modulation DAC must initially be trimmed to achieve the required frequency separation.

The FSK frequency deviation trimming procedure is described in Table 33, page 50.

Table 33 • **Procedure for FSK Deviation Trim**

1. Write **MAN_GLOBAL_EN** = 0x04.	Setting the *man_pll_en* bit turns on the PLL.
2. Wait 2 ms	Wait 2 ms for the PLL to settle before enabling the auto trim.
3. Write **TRIM_CMD** = 0x07.	Writing 0x07 to *trim_cmd* launches the FSK deviation trimming. The **TRIM_CMD** register is cleared internally once the trim procedure is done.
4. Now that the FSK deviation trim procedure is launched: a. Wait for *trim_done_irq*.	 Indicates that the trim has completed. The trim typically takes less than 500 ms.
5. At the end of the FSK Deviation trimming procedure: a. Read **IRQ1**.	 Read to clear the **IRQ1** register and to see whether the *trim_fail_irq* status bit has been set.
6. Write **MAN_GLOBAL_EN** = 0x00.	Disable the PLL.

7.5.10 Output Power Trimming

The transmitter output power trim settings are available for adjusting the power level. A programmable binary code selects the transmitter output power level. See the ZL70550 datasheet for transmit power versus PA trim code.

The transmitter output power trimming procedure is described in Table 34, page 51. Users may determine that the transmitter output power does not need to be trimmed on a part-by-part basis. Instead, a predetermined value can be programmed into every part. If a very accurate output power level is desired, then individual trimming may be necessary. Trim the current reference before any individual trimming is performed.

Table 34 • **Procedure for Transmitter Output Power Trim**

1. Determine desired value ($PA_{desired}$) for transmitter output power level. a. Refer to Figure 7 (Transmit Power vs. PA Trim Value) in the ZL70550 Datasheet. b. Choose the PA trim value associated with the desired TX output power.	See Note 1.
2. Write **PA_PWR_LEVEL[5:0]** = $PA_{desired}$.	Write the desired value (not necessarily the default value) to *pa_pwr_level[5:0]*.
3. Output a carrier. a. Write **MAN_GLOBAL_EN** = 0x01.	
4. Measure the transmitter output power with a power meter or spectrum analyzer.	It is recommended that the PA trim value be selected during the design or manufacturing phase when a power meter or spectrum analyzer can be used to measure the transmitter output power. See Note 1; steps 1 through 4 may need to be repeated to ensure that regulatory standards are met.
5. Store **PA_PWR_LEVEL** value in external nonvolatile memory.	Since the transmitter output power trim is a factory trim, the trim value must be stored in external nonvolatile memory and loaded into the **PA_PWR_LEVEL** register upon initialization of the device. Alternatively, the power can be preset or even dynamically adjusted based on link conditions.

[1] Attention must be paid to the second and third harmonics of the transmitted signal when selecting the PA output power so that regulatory standards are not violated. The transmitter output level can be adjusted such that the standards are not met without a SAW filter or a low-pass filter.

7.5.11 LNA Gain

The LNA gain is controlled by two registers **LNA_GAIN** and **LNA_BIAS_TRIM**. The *lna_bias_trim* field (at **LNA_BIAS_TRIM[5:2]**) acts as the MSBs for the LNA gain. For a current consumption of 2.4 mA, set the **LNA_GAIN** register to 0x0F and leave the **LNA_BIAS_TRIM** register at a default setting of 0x05. To achieve a higher sensitivity set the **LNA_GAIN** register to 0x0F and the **LNA_BIAS_TRIM** register to 0x29. This increases the receiver current consumption to 3.2 mA. Typically, this setting is determined during product development. If an external LNA is used, then lower gain is recommend. This is to prevent overloading the LNA internal to the ZL70550. In the case of the external LNA, the optimum gain setting can be determined by monitoring the sensitivity while increasing the gain. At some gain setting, the sensitivity begins to crown, which means that the noise figure is limiting the sensitivity. At this point, users may consider backing off the gain by one code to reduce the risk of overloading the internal LNA.

8 Registers

8.1 Using the Memory Map

This section contains the address for each register, the bit definitions for the register contents, and some programming notes when appropriate. If not all bits are used, the unused bits are read-only and always return a value of zero. All writable bits can be read back at the same address and bit location as written. For values that are longer than eight bits, multiple register addresses are used and the LSB is in the lowest address register (that is to say, little-endian).

The register bits fall into the following categories.

- Read and Write (R/W). These bits can be written from the control interface and read back.
- Read only (R). These bits are read-only from the control interface and are not cleared on read.
- Clear on Read (CoR). These bits are cleared to zero when read from the control interface.
- Read, Write, Clear on Done (R/W/CoD). These are command bits that are set to start a command. The current state of the bit can be read any time without affecting the bit value. The bit is cleared automatically when the operation of the command is complete.
- Read, Write, Clear on Start (R/W/CoS).These are command bits that are set to start a command. The bit is cleared when the operation starts and therefore always returns 0 on a read operation.

There are three major types of register in the ZL70550 memory map. Although these registers are all the same in hardware terms, they have different logical properties and thus require different application programming methods.

- Bitwise registers

  When writing to a bitwise register, be careful to modify only the targeted bits and to preserve all others. When modifying a register that contains reserved bits, always set the reserved bits to the value given in the register description or to 0 if no value is given.
- Byte-wide registers

  Byte-wise registers can be programmed with a simple write operation.
- Multi-byte registers

  In the ZL70550 memory map, multi-byte registers are little-endian, accessed starting with the least significant byte at the lowest address.

8.2 Address Space

The following table summarizes the ZL70550 memory map and provides recommended initialization values. It is recommended that these values be written when the device is powered on and after every chip reset.

Table 35 • **Memory Map**

Description	Name	Dec Addr	Hex Addr	R/W	Default	Recommended Value[1]
SPI Local Registers *(Address range: decimal 0–7; hexadecimal 0x0–0x7. Addressing mode: short. Always-on: no.)*						
Chip ID	SPIR_CHIP_ID	0	0x0	R	8'h47	
Chip revision	SPIR_REVISION	1	0x1	R	8'h11	
2V reset generator	SPIR_2V_RESET	2	0x2	R/W	8'hCA	
Chip reset	SPIR_SYS_RESET	3	0x3	R/W	8'hBC	
Power down request	SPIR_PWRDWN_REQ	4	0x4	R/W	8'hDE	
SPIR local status	SPIR_LOCAL_STAT	5	0x5	R	7'h00	
SPI control register	SPIR_CTRL	6	0x6	R/W	5'h00	
System Bus Control and Status Registers *(Address range: decimal 256–331; hexadecimal 0x100–0x14B. Addressing mode: long. Always-on: no.)*						
Frame sync control 1	DP_CTRL0	256	0x100	R/W	7'h00	
Received PHY header status	RPHR_STAT	261	0x105	R	8'h08	
RX payload length	RXC_PLD_LEN0	262	0x106	R	8'h00	
	RXC_PLD_LEN1	263	0x107	R	1'h00	
MAC SPI status	MSC_SPI_STAT	264	0x108	R	4'h00	
RX frame length	RXC_FRM_LEN0	265	0x109	R	8'h00	
	RXC_FRM_LEN1	266	0x10A	R	1'h00	
RX frame control for Z-Star packet mode	RXC_FRM_CTRL0	267	0x10B	R	8'h00	
	RXC_FRM_CTRL1	268	0x10C	R	8'h00	
RX source short ID	RXC_SOURCE_SID	269	0x10D	R	8'h00	
RX network ID	RXC_NETWORK_ID	270	0x10E	R	8'h00	
RX destination short ID	RXC_DEST_SID	271	0x10F	R	8'h00	

1. Use this recommended initial register setting by writing this value after every chip reset. Device characterization has shown that the recommended value gives better system performance.
2. For Z-Star operation, use recommended values in Section 6.5.1 Z-Star Operation, page 38.
3. Recommended value may vary based on customer configuration. Refer to Sections 9.3 SPI Local Registers, page 61, 9.4 System Bus Control and Status Registers, page 63, and 9.5 Always-On System Bus Control and Status Registers, page 75.
4. Required value. Other values may cause errors for some configurations.
5. It is recommended that the default value (or reset value) be used.
6. Do not write to this register.

Table 35 • **Memory Map *(continued)***

Description	Name	Dec Addr	Hex Addr	R/W	Default	Recommended Value[1]
RX destination long ID	RXC_DEST_LID0	272	0x110	R	8'h00	
	RXC_DEST_LID1	273	0x111	R	8'h00	
	RXC_DEST_LID2	274	0x112	R	8'h00	
	RXC_DEST_LID3	275	0x113	R	8'h00	
	RXC_DEST_LID4	276	0x114	R	8'h00	
	RXC_DEST_LID5	277	0x115	R	8'h00	
	RXC_DEST_LID6	278	0x116	R	8'h00	
	RXC_DEST_LID7	279	0x117	R	8'h00	
RX source long ID	RXC_SOURCE_LID0	280	0x118	R	8'h00	
	RXC_SOURCE_LID1	281	0x119	R	8'h00	
	RXC_SOURCE_LID2	282	0x11A	R	8'h00	
	RXC_SOURCE_LID3	283	0x11B	R	8'h00	
	RXC_SOURCE_LID4	284	0x11C	R	8'h00	
	RXC_SOURCE_LID5	285	0x11D	R	8'h00	
	RXC_SOURCE_LID6	286	0x11E	R	8'h00	
	RXC_SOURCE_LID7	287	0x11F	R	8'h00	
MAC RX frame status	RXC_FRM_STAT0	288	0x120	R	7'h00	
	RXC_FRM_STAT1	289	0x121	R	7'h00	
MAC sequencer status	MSC_FRM_STAT0	290	0x122	R	8'h00	
	MSC_FRM_STAT1	291	0x123	R	8'h00	
Manual global enables	MAN_GLOBAL_EN	294	0x126	R/W	3'h00	
Manual miscellaneous	MAN_TEST	302	0x12E	R/W	4'h04	
Clock enable tests	CLK_TEST	305	0x131	R/W	8'h00	
ADC mode conversion (start single ADC average conversion mode and continuous ADC average conversion mode)	ADC_CONV_START	314	0x13A	R/W	2'h00	
Maximum result in ADC modes and when using RSSI in CSMA-CA modes	ADC_MAX	315	0x13B	R	7'h00	
Average result in ADC modes and when using RSSI in CSMA-CA modes	ADC_AVG	316	0x13C	R	7'h00	

1. Use this recommended initial register setting by writing this value after every chip reset. Device characterization has shown that the recommended value gives better system performance.
2. For Z-Star operation, use recommended values in Section 6.5.1 Z-Star Operation, page 38.
3. Recommended value may vary based on customer configuration. Refer to Sections 9.3 SPI Local Registers, page 61, 9.4 System Bus Control and Status Registers, page 63, and 9.5 Always-On System Bus Control and Status Registers, page 75.
4. Required value. Other values may cause errors for some configurations.
5. It is recommended that the default value (or reset value) be used.
6. Do not write to this register.

Table 35 • **Memory Map *(continued)***

Description	Name	Dec Addr	Hex Addr	R/W	Default	Recommended Value[1]
Trimmed RSSI average at the completion of the DC restore process in RSSI detect modes with CSMA not active	ADC_AVG_TRMD_PKT_RSSI	319	0x13F	R	7'h00	
IRQ status	IRQ0	323	0x143	CoR	3'h00	
	IRQ1	324	0x144	CoR	8'h00	
	IRQ2	325	0x145	CoR	8'h00	
Trim command	TRIM_CMD	327	0x147	R/W	4'h00	
LNA trim peak ADC value	LNA_PEAK_ADC	330	0x14A	R	5'h00	
Antenna trim peak ADC value	ANT_PEAK_ADC	331	0x14B	R	5'h00	Note 5
Always-On System Bus Control and Status Registers *(Address range: decimal 512–671; hexadecimal 0x200–0x29F. Addressing mode: long. Always-on: yes.)*						
VCO frequency tune value for the RX mode	VCO_FRQ_RX_TRIM_L	513	0x201	R/W	8'h88	
	VCO_FRQ_RX_TRIM_H	514	0x202	R/W	3'h00	
VCO frequency tune value for the TX mode (with modulation off) with the power amplifier off (only bias on)	VCO_FRQ_TXPAOFF_TRIM_L	515	0x203	R/W	8'h97	
	VCO_FRQ_TXPAOFF_TRIM_H	516	0x204	R/W	3'h00	
VCO frequency tune value for the TX mode (with modulation off) with the power amplifier on	VCO_FRQ_TXPAON_TRIM_L	517	0x205	R/W	8'h98	
	VCO_FRQ_TXPAON_TRIM_H	518	0x206	R/W	3'h00	
VCO frequency tune MSB value at output of multiplexer	VCO_FRQ_TRIM_L	519	0x207	R	8'hFF	
	VCO_FRQ_TRIM_H	520	0x208	R	3'h04	
VCO frequency band trim value	VCO_FRQ_BAND_TRIM	521	0x209	R/W	4'h07	
Modulator DAC trim value	MOD_DAC_TRIM	522	0x20A	R/W	5'h00	
Frequency deviation trim target	FSK_DEV_TRIM_TARGET	524	0x20C	R/W	8'h66	US: 8’h6F, EU: 8’h76, China: 8’h83 (Note 3)
RC oscillator frequency trim value	RCOSC_FREQ_TRIM	525	0x20D	R/W	6'h00	

1. Use this recommended initial register setting by writing this value after every chip reset. Device characterization has shown that the recommended value gives better system performance.
2. For Z-Star operation, use recommended values in Section 6.5.1 Z-Star Operation, page 38.
3. Recommended value may vary based on customer configuration. Refer to Sections 9.3 SPI Local Registers, page 61, 9.4 System Bus Control and Status Registers, page 63, and 9.5 Always-On System Bus Control and Status Registers, page 75.
4. Required value. Other values may cause errors for some configurations.
5. It is recommended that the default value (or reset value) be used.
6. Do not write to this register.

Table 35 • **Memory Map *(continued)***

Description	Name	Dec Addr	Hex Addr	R/W	Default	Recommended Value[1]
Manual LNA frequency trim	LNA_FRQ_TRIM	526	0x20E	R/W	5'h16	
Manual LNA trim value	LNA_BIAS_TRIM	527	0x20F	R/W	6'h05	Note 3
IREF resistor trim value	IREF_TRIM	528	0x210	R/W	5'h0F	
Crystal oscillator trim value	XO_TRIM	529	0x211	R/W	6'h26	
Gaussian filter trim value	GAUS_TRIM	532	0x214	R/W	8'h62	
VCO amplitude trim value	VCO_AMP_TRIM	533	0x215	R/W	6'h3F	
Antenna trim value	ANT_TRIM	534	0x216	R/W	5'h16	
RX peak detector trim value	RX_PKDET_TRIM	535	0x217	R/W	5'h09	
Initial DC correct/AFC value	DC_CNTR_TRIM	536	0x218	R/W	8'h62	
IRQ enables	IRQ_EN0	537	0x219	R/W	4'h08	
	IRQ_EN1	538	0x21A	R/W	8'h00	
	IRQ_EN2	539	0x21B	R/W	8'h00	
Pad enables	PAD_EN0	540	0x21C	R/W	8'h00	
TX control	TX_CTRL0	542	0x21E	R/W	8'h1B	
	TX_CTRL1	543	0x21F	R/W	7'h01	
	TX_CTRL2	544	0x220	R/W	8'hA2	
TX frame-packet buffer length	TX_BUF_LEN0	545	0x221	R/W	8'h00	
	TX_BUF_LEN1	546	0x222	R/W	1'h00	
TX frame control for Z-Star packet mode	TX_FRM_CTRL	547	0x223	R/W	8'h01	
TX frame sequence number for Z-Star packet mode	TX_FRM_SEQ_NO	548	0x224	R/W	4'h00	
RX control for Z-Star packet mode	RX_CTRL0	549	0x225	R/W	8'h0B	Note 2
	RX_CTRL1	550	0x226	R/W	8'h12	Note 2
	RX_CTRL2	551	0x227	R/W	8'h84	
Address mask	ADDR_MASK0	553	0x229	R/W	8'hFF	Note 2
	ADDR_MASK1	554	0x22A	R/W	8'hFF	Note 2
RX buffer length	RX_FRM_LEN0	555	0x22B	R/W	8'h00	
	RX_FRM_LEN1	556	0x22C	R/W	1'h01	

1. Use this recommended initial register setting by writing this value after every chip reset. Device characterization has shown that the recommended value gives better system performance.
2. For Z-Star operation, use recommended values in Section 6.5.1 Z-Star Operation, page 38.
3. Recommended value may vary based on customer configuration. Refer to Sections 9.3 SPI Local Registers, page 61, 9.4 System Bus Control and Status Registers, page 63, and 9.5 Always-On System Bus Control and Status Registers, page 75.
4. Required value. Other values may cause errors for some configurations.
5. It is recommended that the default value (or reset value) be used.
6. Do not write to this register.

Table 35 • **Memory Map *(continued)***

Description	Name	Dec Addr	Hex Addr	R/W	Default	Recommended Value[1]
RX buffer write threshold	RX_HDR_THRESH	557	0x22D	R/W	8'h00	
Network ID	NETWORK_ID	558	0x22E	R/W	8'h00	
Device short ID: address of this device	DEVICE_SID	559	0x22F	R/W	8'h00	
Short ID: address of other device	OTHER_SID	560	0x230	R/W	8'h00	
Long ID of this device	DEVICE_LID0	561	0x231	R/W	8'h00	
	DEVICE_LID1	562	0x232	R/W	8'h00	
	DEVICE_LID2	563	0x233	R/W	8'h00	
	DEVICE_LID3	564	0x234	R/W	8'h00	
	DEVICE_LID4	565	0x235	R/W	8'h00	
	DEVICE_LID5	566	0x236	R/W	8'h00	
	DEVICE_LID6	567	0x237	R/W	8'h00	
	DEVICE_LID7	568	0x238	R/W	8'h00	
Long ID of other device	OTHER_LID0	569	0x239	R/W	8'h00	
	OTHER_LID1	570	0x23A	R/W	8'h00	
	OTHER_LID2	571	0x23B	R/W	8'h00	
	OTHER_LID3	572	0x23C	R/W	8'h00	
	OTHER_LID4	573	0x23D	R/W	8'h00	
	OTHER_LID5	574	0x23E	R/W	8'h00	
	OTHER_LID6	575	0x23F	R/W	8'h00	
	OTHER_LID7	576	0x240	R/W	8'h00	
MAC controls	MAC_CTRL	577	0x241	R/W	5'h00	Notes 2, 5
CRC polynomial	CRC_POLY0	578	0x242	R/W	8'h01	
	CRC_POLY1	579	0x243	R/W	8'h8F	
	CRC_POLY2	580	0x244	R/W	8'h00	
	CRC_POLY3	581	0x245	R/W	8'h00	
SPI control reg	SPI_CTRL	586	0x24A	R/W	4'h07	Note 2
CSMA control	CSMA_CTRL	587	0x24B	R/W	7'h45	
CSMA 100-µs counts per interval	CSMA_TIME0	588	0x24C	R/W	8'h0A	
	CSMA_TIME1	589	0x24D	R/W	4'h00	

1. Use this recommended initial register setting by writing this value after every chip reset. Device characterization has shown that the recommended value gives better system performance.
2. For Z-Star operation, use recommended values in Section 6.5.1 Z-Star Operation, page 38.
3. Recommended value may vary based on customer configuration. Refer to Sections 9.3 SPI Local Registers, page 61, 9.4 System Bus Control and Status Registers, page 63, and 9.5 Always-On System Bus Control and Status Registers, page 75.
4. Required value. Other values may cause errors for some configurations.
5. It is recommended that the default value (or reset value) be used.
6. Do not write to this register.

Table 35 • **Memory Map *(continued)***

Description	Name	Dec Addr	Hex Addr	R/W	Default	Recommended Value[1]
CSMA maximum random number of intervals in the back-off	CSMA_MAX_RAND_BACKOFF0	590	0x24E	R/W	8'h32	
	CSMA_MAX_RAND_BACKOFF1	591	0x24F	R/W	4'h00	
CSMA retry minimum back-off in intervals	CSMA_RETRY_MIN_BACKOFF0	592	0x250	R/W	8'h19	
	CSMA_RETRY_MIN_BACKOFF1	593	0x251	R/W	4'h00	
Synchronization threshold minimum number of bits that need to match the 40-bit sync word	SYNC_THRESH	594	0x252	R/W	6'h24	6’h15
Preamble length in bytes used on TX side	PREAM_LEN	595	0x253	R/W	8'h1A	8’h0A
Delay time count for ***prx_tx_trig***	TX_TRIG_CNT	596	0x254	R/W	8'h14	
Clock recovery fast adjust mode error threshold	ADJ_FAST	597	0x255	R/W	8'h58	8’h25
Clock recovery slow adjust mode error threshold	ADJ_SLOW	598	0x256	R/W	8'h58	8’h25
Length of DC adjustment in bytes	PREAM_LOCK_CNT	608	0x260	R/W	8'h12	8’h07
Selection for GPIO0	GPIO_0_SEL	609	0x261	R/W	8'h00	
Selection for GPIO1	GPIO_1_SEL	610	0x262	R/W	8'h00	
Selection for GPIO2	GPIO_2_SEL	611	0x263	R/W	8'h00	
Selection for GPIO3	GPIO_3_SEL	612	0x264	R/W	8'h00	
Frame sync pattern	SYNC_PTRN0	613	0x265	R/W	8'h13	Note 5
	SYNC_PTRN1	614	0x266	R/W	8'hDA	Note 5
	SYNC_PTRN2	615	0x267	R/W	8'h32	Note 5
	SYNC_PTRN3	616	0x268	R/W	8'hEC	Note 5
	SYNC_PTRN4	617	0x269	R/W	8'h79	Note 5
PLL start-up delay (100-µs count)	PLL_START_DLY	618	0x26A	R/W	8'h14	
TX PLL turn-around settling delay count (10-µs count)	TX_PLL_DLY_CNT	619	0x26B	R/W	8'h1E	8’h0F
RX PLL turn-around settling delay count (10-µs count)	RX_PLL_DLY_CNT	620	0x26C	R/W	8'h26	8’h16

1. Use this recommended initial register setting by writing this value after every chip reset. Device characterization has shown that the recommended value gives better system performance.
2. For Z-Star operation, use recommended values in Section 6.5.1 Z-Star Operation, page 38.
3. Recommended value may vary based on customer configuration. Refer to Sections 9.3 SPI Local Registers, page 61, 9.4 System Bus Control and Status Registers, page 63, and 9.5 Always-On System Bus Control and Status Registers, page 75.
4. Required value. Other values may cause errors for some configurations.
5. It is recommended that the default value (or reset value) be used.
6. Do not write to this register.

Table 35 • **Memory Map *(continued)***

Description	Name	Dec Addr	Hex Addr	R/W	Default	Recommended Value[1]
CW count	CW_DLY_CNT	621	0x26D	R/W	8'h26	8’h05
PLL fast to slow loop delay count	SLO_DLY_CNT	622	0x26E	R/W	8'h44	8’hFF
PA ramp-down delay count at the end of the TX data (10-µs count)	PA_OFF_DLY_CNT	623	0x26F	R/W	4'h03	4’h00 (Note 4)
RX time limit for acknowledgment timeout (100-µs count)	ACK_TIME_LIMIT	624	0x270	R/W	8'h18	8’h0C (Note 3)
RX time limit for packet timeout (100-µs count)	PKT_TIME_LIMIT	625	0x271	R/W	8'h24	8’h0C (Note 3)
RX frame sequence number	RX_FRM_SEQ_NO	626	0x272	R/W	4'h00	
TX nonacknowledgment packet transmitted count	TX_PKT_CNT	627	0x273	WCOR	8'h00	
TX packet retry accumulated count (after ACK failure)	TX_PKT_RETRY_CNT	628	0x274	WCOR	8'h00	
TX packet drop accumulated count (after ACK failure)	TX_PKT_DROP_CNT	629	0x275	WCOR	8'h00	
RX nonacknowledgment packet received count	RX_PKT_CNT	630	0x276	WCOR	8'h00	
RX packet received count for all types	RX_ALL_PKT_CNT	631	0x277	WCOR	8'h00	
RX nonacknowledgment packet drop accumulated count	RX_PKT_DROP_CNT	632	0x278	WCOR	8'h00	
RX packet sync error accumulated count	RX_SYNC_ERR_CNT	633	0x279	WCOR	8'h00	
RX packet all error accumulated count for all errors	RX_ALL_ERR_CNT	634	0x27A	WCOR	8'h00	
CSMA retry count	CSMA_RETRY_CNT	635	0x27B	WCOR	8'h00	
CSMA fail count	CSMA_FAIL_CNT	636	0x27C	WCOR	8'h00	
System bus clock ***cg_sys_clk*** divide count	SYS_CLK_DIV	637	0x27D	R/W	5'h16	5’h14
PLL clock divide count	PLL_CLK_DIV_CNT	638	0x27E	R/W	7'h51	7’h50

1. Use this recommended initial register setting by writing this value after every chip reset. Device characterization has shown that the recommended value gives better system performance.
2. For Z-Star operation, use recommended values in Section 6.5.1 Z-Star Operation, page 38.
3. Recommended value may vary based on customer configuration. Refer to Sections 9.3 SPI Local Registers, page 61, 9.4 System Bus Control and Status Registers, page 63, and 9.5 Always-On System Bus Control and Status Registers, page 75.
4. Required value. Other values may cause errors for some configurations.
5. It is recommended that the default value (or reset value) be used.
6. Do not write to this register.

Table 35 • **Memory Map *(continued)***

Description	Name	Dec Addr	Hex Addr	R/W	Default	Recommended Value[1]
PHY RX mode select	PHY_RX_MODE_SEL	642	0x282	R/W	8'h45	0 for bit [6]
Frame sync timeout delay (in bytes)	SYNC_DLY_CNT	643	0x283	R/W	8'h22	8’h0F
Preamble detect threshold and bandwidth	PREAM_DET_CTRL	644	0x284	R/W	8'h6B	8’h33
Frame sync control 2	DPORT_CTRL	647	0x287	R/W	6'h1D	6’h1B (Note 3)
PHY TX raw mode control	PHY_TX_RAW_MODE_CTRL	648	0x288	R/W	2'h00	
Data rate control	RATE_CTRL	649	0x289	R/W	8'h00	
M divide counter value	SYNTH_M_DIV	650	0x28A	R/W	8'hB5	
A divide counter value	SYNTH_A_DIV	651	0x28B	R/W	6'h07	
ADC modes multiplexer input selection	ADC_MUX_IN_SEL	654	0x28E	R/W	3'h00	
Number of conversions used for averaging in ADC modes and when using RSSI in CSMA-CA modes	ADC_POW_N_CONV	655	0x28F	R/W	7'h30	
ADC RSSI threshold	ADC_RSSI_THRESH	659	0x293	R/W	7'h00	7'h01
ADC CSMA threshold	ADC_CSMA_THRESH	660	0x294	R/W	7'h20	
LNA gain	LNA_GAIN	661	0x295	R/W	4'h07	4’h0F
Analog control	ANA_CTRL0	663	0x297	R/W	8'h80	8’h84
Lock detector and pump select	LOCK_PUMP_SEL	665	0x299	R/W	6'h08	
PA power level	PA_PWR_LEVEL	666	0x29A	R/W	8'h88	
PA buffer bias control	VCO_BUF_BIAS	667	0x29B	R/W	4'h03	Note 5
VCO control	VCO_CTRL	668	0x29C	R/W	7'h08	
VCO control voltage comparator reference values	VCO_CMP_VREF_CTRL	670	0x29E	R/W	2'h00	2’h01

System Bus Transmit Buffer (SRAM)
(Address range: decimal 1024–1535; hexadecimal 0x400–0x5FF. Addressing mode: long. Always-on: no.)

System Bus Receive Buffer (SRAM)
(Address range: decimal 1536–2047; hexadecimal 0x600–0x7FF. Addressing mode: long. Always-on: no.)

1. Use this recommended initial register setting by writing this value after every chip reset. Device characterization has shown that the recommended value gives better system performance.
2. For Z-Star operation, use recommended values in Section 6.5.1 Z-Star Operation, page 38.
3. Recommended value may vary based on customer configuration. Refer to Sections 9.3 SPI Local Registers, page 61, 9.4 System Bus Control and Status Registers, page 63, and 9.5 Always-On System Bus Control and Status Registers, page 75.
4. Required value. Other values may cause errors for some configurations.
5. It is recommended that the default value (or reset value) be used.
6. Do not write to this register.

9.3 SPI Local Registers

Addresses 0x000 to 0x007 comprise the SPI local registers, which use short addressing mode.

These registers are not powered from VSUP (that is, they are volatile registers rather than always-on registers). Whenever the ZL70550 device goes into its SLEEP state, information contained in these registers is lost. All necessary registers must therefore be backed up external to the MAC before the ZL70550 device goes to SLEEP, and the registers must be reloaded on wake-up.

9.3.1 Chip ID

Short Name: SPIR_CHIP_ID
Address: 0x0
Always-On: No

Table 36 • **SPIR_CHIP_ID**

Field Name	Bit	Access	Description	Default
spir_chip_id	7:0	WCOR	Chip ID. This eight-bit value gives the ID of the chip.	0x47

9.3.2 Chip Revision

Short Name: SPIR_REVISION
Address: 0x1
Always-On: No

Table 37 • **SPIR_REVISION**

Field Name	Bit	Access	Description	Default
spir_revision	7:0	R	Chip revision. This eight-bit value gives the revision of the current implementation.	0x11

9.3.3 2V Reset Generator

Short Name: SPIR_2V_RESET
Address: 0x2
Always-On: No

Table 38 • **SPIR_2V_RESET**

Field Name	Bit	Access	Description	Default
spir_2v_reset	7:0	R/W	VDDD S/W reset. Write reset value (0xCA) to perform a VDDD reset. Writing 0xCA causes a hardware reset to the VDDD section of the IC, resulting in the assertion of ***cg_sys_rst***.	0xCA

9.3.4 Chip Reset

Short Name: SPIR_SYS_RESET
Address: 0x3
Always-On: No

Table 39 • **SPIR_SYS_RESET**

Field Name	Bit	Access	Description	Default
spir_sys_reset	7:0	R/W	Hardware reset, similar to pad reset. Write 0xBC to perform a hardware reset. Writing 0xBC causes a hardware reset to the VDDD section of the IC, similar to resetting the device via the **RESET_B** pin. This results in resetting the entire chip and turning off VDDD.	0xBC

9.3.5 Power Down Request

Short Name: SPIR_PWRDWN_REQ
Address: 0x4
Always-On: No

Table 40 • **SPIR_PWRDWN_REQ**

Field Name	Bit	Access	Description	Default
spir_pwrdwn_req	7:0	R/W	Power-down request to the 3-volt power-down sequencer. Write 0xDE to perform a VDDD (2-volt) power-down. This puts the device into the SLEEP state.	0xDE

9.3.6 SPIR Local Status

Short Name: SPIR_LOCAL_STAT
Address: 0x5
Always-On: No

Table 41 • **SPIR_LOCAL_STAT**

Field Name	Bit	Access	Description	Default
–	7	R	<Reserved>	0x0
dp_pll_lock_2q	6	R	PLL lock status	0x0
mac_pll_en	5	R	MAC enable status; also indicates if PLL is enabled. Reflects the status of *msc_blk_en* at MAC_CTRL[1].	0x0
msc_synth_rdy	4	R	Synthesizer ready status	0x0
spir_xtal_osc_off	3	R	Crystal oscillator status; is high when crystal oscillator is turned off	0x0
spis_rx_cmd	2	R	SPI receive command active status; is high when a SPI RX command is active	0x0
spis_tx_req_cmd	1	R	SPI data request packet command active status; is high when a SPI TX request command is active	0x0
spis_tx_cmd	0	R	SPI transmit command active status; is high when a SPI TX command is active	0x0

9.3.7 SPI Control Register

Short Name: SPIR_CTRL
Address: 0x6
Always-On: No

Table 42 • **SPIR_CTRL**

Field Name	Bit	Access	Description	Default
–	7:3	R/W	<Reserved; always write 0x00 (or 5’b00000) to these bits>	0x00
spir_addr_incr_inh	2	R/W	Inhibit address autoincrement for multibyte SPI operations. If set high, all bytes are read from or written to the same address in the command.	0x0
–	1:0	R/W	<Reserved; always write 0x0 (2’b00) or to these bits>	0x0

9.4 System Bus Control and Status Registers

Addresses 0x100 to 0x14B comprise the system bus control and status registers, which use long addressing mode.

These registers are not powered from VSUP (that is, they are volatile rather than always-on registers). Whenever the ZL70550 device goes into its SLEEP state, information contained in these registers is lost. All necessary registers must therefore be backed up external to the MAC before the ZL70550 device goes to SLEEP, and the registers must be reloaded on wake-up.

9.4.1 Frame Sync Control 1

Short Names: DP_CTRL0
Addresses: 0x100
Always-On: No

Table 43 • **DP_CTRL0**

Field Name	Bit	Access	Description	Default
–	7:3	R/W	<Reserved; always write 0x00 (or 5’b00000) to these bits>	0x00
dp_sync_always	2	R/W	Send preamble once, followed continuously by frame sync pattern. Used in raw bit mode.	0x0
–	1:0	R/W	<Reserved; always write 0x0 (or 2’b00) to these bits>	0x0

9.4.2 Received PHY Header Status

Short Name: RPHR_STAT
Address: 0x105
Always-On: No

Table 44 • **RPHR_STAT**

Field Name	Bit	Access	Description	Default
prx_rx_rate	7:6	R	Detected RX bit rate. Also used for transmit rate if *tx_follow_rx_rate* is 1. • 00: 200 kbit/s • 01: 100 kbit/s • 10: 50 kbit/s	0x0
cdr_inverted_sync	5	R	Inverted sync detected, cleared on next sync detect. Used to enable FEC on RX, and FEC and inverted frame sync pattern on TX.	0x0
rphr_fcs_len	4:2	R	FCS length in bytes, up to four bytes: • 001: One byte (not valid for Z-Star packet mode) • 010: Two bytes • 011: Three bytes (not valid for Z-Star packet mode) • 100: Four bytes In Z-Star packet mode, this is the value in the received PHY header. In user and raw packet modes, this is copied from *rx_fcs_len*.	0x2
rphr_frm_format	1:0	R	PHY header frame format, included in received PHY header in Z-Star packet mode: • 00: Two bytes • Others: not currently supported	0x0

9.4.3 RX Payload Length

Short Name: RXC_PLD_LEN (comprising RXC_PLD_LEN0 and RXC_PLD_LEN1)
Addresses: Two-byte little-endian starting at 0x106 (comprising 0x106 and 0x107, respectively)
Always-On: No

Table 45 • **RXC_PLD_LEN**

Field Name	Bit	Access	Description	Default
–	15:9	R	<Reserved>	0x00
rxc_pld_len	8:0	R	Length in bytes of received payload without headers and FCS. In all packet modes, if *wr_rx_payld_only* is set to 1, then the SPI status information returned to the host on MISO includes *rxc_pld_len* as ***rxc_pkt_buf_len[8:0]*** (see Table 10, page 27).	0x000

9.4.4 MAC SPI Status

Short Name: MSC_SPI_STAT
Address: 0x108
Always-On: No

Table 46 • **MSC_SPI_STAT**

Field Name	Bit	Access	Description	Default
–	7:4	R	<Reserved>	0x0
msc_cmd_busy	3	R	MAC sequencer busy with active SPI command	0x0
spis_rx_cmd	2	R	SPI RX command active	0x0
spis_tx_req_cmd	1	R	SPI TX data request command active	0x0
spis_tx_cmd	0	R	SPI TX command active	0x0

9.4.5 RX Frame Length

Short Name: RXC_FRM_LEN (comprising RXC_FRM_LEN0 and RXC_FRM_LEN1)
Addresses: Two-byte little-endian starting at 0x109 (comprising 0x109 and 0x10A, respectively)
Always-On: No

Table 47 • **RXC_FRM_LEN**

Field Name	Bit	Access	Description	Default
–	15:9	R	<Reserved>	0x00
rxc_frm_len	8:0	R	Length in bytes of received PHY frame, including headers and FCS. Included in PHY header for Z-Star and user packet formats. In all packet modes, if *wr_rx_payld_only* is set to 0, then the SPI status information returned to the host on MISO includes *rxc_frm_len* as ***rxc_pkt_buf_len[8:0]*** (see Table 10, page 27).	0x000

9.4.6 Fields Received in Z-Star MAC Header

9.4.6.1 RX Frame Control for Z-Star Packet Mode

Short Name: RXC_FRM_CTRL0 and RXC_FRM_CTRL1
Address: 0x10B and 0x10C, respectively
Always-On: No

Table 48 • **RXC_FRM_CTRL0**

Field Name	Bit	Access	Description	Default
rxc_addr_mode	7	R	RX address mode from RX header: • 0: Short addressing mode • 1: Long addressing mode	0x0

Table 48 • **RXC_FRM_CTRL0** ***(continued)***

Field Name	Bit	Access	Description	Default
rxc_rx_reserved[0]	6	R	Reserved bit from RX header (always 0)	0x0
rxc_frm_stype	5:3	R	RX frame subtype from RX header; refer to Table 97, page 85, for "Frame Type - Frame Subtype" definition in description of field *tx_frm_type*	0x0
rxc_frm_type	2:0	R	RX frame type from RX header; refer to Table 97, page 85, for "Frame Type - Frame Subtype" definition in description of field *tx_frm_type.*	0x0

Table 49 • **RXC_FRM_CTRL1**

Field Name	Bit	Access	Description	Default
rxc_secure_en	7	R	Security indicator from RX header; payload is encrypted if set to 1	0x0
rxc_frm_pend	6	R	RX frame pending status from RX header	0x0
rxc_ack_req	5	R	RX acknowledgment request from RX header	0x0
rxc_rx_reserved[1]	4	R	Reserved bit in RX header (always 0)	0x0
rxc_frm_seq_no	3:0	R	RX frame sequence number from RX header	0x0

9.4.6.2 RX Source Short ID

Short Name: RXC_SOURCE_SID
Address: 0x10D
Always-On: No

Table 50 • **RXC_SOURCE_SID**

Field Name	Bit	Access	Description	Default
rxc_source_sid	7:0	R	RX source short ID from RX header	0x00

9.4.6.3 RX Network ID

Short Name: RXC_NETWORK_ID
Address: 0x10E
Always-On: No

Table 51 • **RXC_NETWORK_ID**

Field Name	Bit	Access	Description	Default
rxc_network_id	7:0	R	RX network ID from RX header	0x00

9.4.6.4 RX Destination Short ID

Short Name: RXC_DEST_SID
Address: 0x10F
Always-On: No

Table 52 • **RXC_DEST_SID**

Field Name	Bit	Access	Description	Default
rxc_dest_sid	7:0	R	RX destination short ID from RX header	0x00

9.4.6.5 RX Destination Long ID

Short Name: RXC_DEST_LID (comprising RXC_DEST_LID0, RXC_DEST_LID1, RXC_DEST_LID2, RXC_DEST_LID3, RXC_DEST_LID4, RXC_DEST_LID5, RXC_DEST_LID6, and RXC_DEST_LID7)
Addresses: Eight-byte little-endian starting at 0x110 (comprising 0x110, 0x111, 0x112, 0x113, 0x114, 0x115, 0x116, and 0x117, respectively)
Always-On: No

Table 53 • **RXC_DEST_LID**

Field Name	Bit	Access	Description	Default
rxc_dest_lid	63:0	R	RX destination long ID from RX header	0x0000 0000 0000 0000

9.4.6.6 RX Source Long ID

Short Name: RXC_SOURCE_LID (comprising RXC_SOURCE_LID0, RXC_SOURCE_LID1, RXC_SOURCE_LID2, RXC_SOURCE_LID3, RXC_SOURCE_LID4, RXC_SOURCE_LID5, RXC_SOURCE_LID6, and RXC_SOURCE_LID7)
Addresses: Eight-byte little-endian starting at 0x118 (comprising0x118, 0x119, 0x11A, 0x11B, 0x11C, 0x11D, 0x11E, and 0x11F, respectively)
Always-On: No

Table 54 • **RXC_SOURCE_LID**

Field Name	Bit	Access	Description	Default
rxc_source_lid	63:0	R	RX source long ID from RX header	0x0000 0000 0000 0000

9.4.7 MAC RX Frame Status

Short Name: RXC_FRM_STAT0 and RXC_FRM_STAT1
Addresses: 0x120 and 0x121 (respectively)
Always-On: No

Table 55 • **RXC_FRM_STAT0**

Field Name	Bit	Access	Description	Default
–	7	R	<Reserved>	0x0
rxc_timeout	6	R	Reception timeout: bad frame sync or no response (dropped RX packet)	0x0
rxc_pcrc_fail	5	R	PHY CRC reception failed (dropped RX packet)	0x0
rxc_mcrc_fail	4	R	MAC CRC reception failed (dropped RX packet)	0x0
rxc_addr_fail	3	R	Address reception failed (dropped RX packet)	0x0
rxc_pkt_fail	2	R	Packet reception failed (dropped RX packet)	0x0
rxc_pkt_pass	1	R	Good packet received status	0x0
rxc_rssi_rx_abort	0	R	RX aborted because RSSI went low after frame sync detection	0x0

Table 56 • **RXC_FRM_STAT1**

Field Name	Bit	Access	Description	Default
–	7	R	<Reserved>	0x0
msc_rx_null_data_frm	6	R	RX null data frame received	0x0
msc_rx_data_frm	5	R	Received RX frame that is neither an acknowledgment, data request, nor null frame	0x0
msc_rx_data_req_frm	4	R	RX data request type frame received	0x0
msc_rx_ack_frm	3	R	RX acknowledgment type frame received	0x0
–	2	R	<Reserved>	0x0
rxc_brdcst_match	1	R	Broadcast address match on received RX frame.This applies to both long and short broadcast mode: • In long addressing mode, the network ID is not used for matching. This bit is set if long addressing mode is used (*rxc_addr_mode* equals 1), **and** broadcast is enabled (*lng_brdcst_en* equals 1), **and** the destination address is all 1s. • In short addressing mode, the unmasked portion of network ID must also match. This bit is set if short addressing mode is used (*rxc_addr_mode* equals 0), **and** broadcast is enabled (*shrt_brdcst_en* equals 1), **and** the destination address is all 1s, **and** the unmasked portion of network ID matches.	0x0
rxc_addr_match	0	R	Address match on received RX frame. Is automatically set if **all** of the following are true: • The destination ID in the received header (*rxc_dest_sid* or *rxc_dest_lid*) matches local device ID (*device_sid* or *device_lid*) • **AND** the source ID in the received header (*rxc_source_sid* or *rxc_source_lid*) matches the unmasked portion of the other ID (*other_sid* or *other_lid*) • **AND** the network ID in the received header (*rxc_network_id)* matches the unmasked portion of the network ID *(network_id*).	0x0

9.4.8 MAC Sequencer Status

Short Name: MSC_FRM_STAT0 and MSC_FRM_STAT1
Addresses: 0x122 and 0x123 (respectively)
Always-On: No

Table 57 • **MSC_FRM_STAT0**

Field Name	Bit	Access	Description	Default
txc_tx_frm_pend	7	R	TX frame pending was set in transmitted TX frame	0x0
txc_tx_ack_req	6	R	TX acknowledgment request was set in transmitted TX frame	0x0
–	5:0	R	<Reserved>	0x00

Table 58 • **MSC_FRM_STAT1**

Field Name	Bit	Access	Description	Default
msc_rx_dup_frm	7	R	MAC RX duplicate frame received	0x0
msc_rx_frm_type_fail	6	R	RX frame type error: RX acknowledgment/packet error, wrong frame type received in header	0x0
msc_csma_fail	5	R	CSMA failure: no transmit	0x0
msc_tx_ack_fail	4	R	TX acknowledgment failure	0x0
msc_tx_pkt_fail	3	R	TX nonacknowledgment packet failure	0x0
msc_txreq_fail	2	R	TX data request failure	0x0
msc_rx_pkt_fail	1	R	RX nonacknowledgment packet failure	0x0
msc_rx_ack_fail	0	R	RX acknowledgment error	0x0

9.4.9 Manual Global Enables

Short Name: MAN_GLOBAL_EN
Address: 0x126
Always-On: No

Table 59 • **MAN_GLOBAL_EN**

Field Name	Bit	Access	Description	Default
–	7:3	R	<Reserved>	0x00
man_pll_en	2	R/W	Enable PLL in manual mode	0x0
rfmac_rcvr_en	1	R/W	Enable receive channel in manual mode	0x0
rfmac_xmtr_en	0	R/W	Enable transmit channel in manual mode	0x0

1. *man_pll_en* enables the following signals: presc_en, pa_en, pll_clk_en, pfd_en, pump_en, lock_en, vco_en, gaus_en, iref_const_en, vrefva_bgap_en, vrefva_buf_en1, and vrefva_buf_en0.
2. *rfmac_rcvr_en* enables the following signals: rf_en, rx_pkdet_en, if_filt_en, limit_en, rssi_en, fm_det_en, and all signals enabled by *man_pll_en*.
3. *rfmac_xmtr_en* enables the following signals: mod_dac_en, ***pa_tx_en***, and all signals enabled by *man_pll_en*.

9.4.10 Manual Miscellaneous

Short Name: MAN_TEST
Address: 0x12E
Always-On: No

Table 60 • **MAN_TEST**

Field Name	Bit	Access	Description	Default
–	7:3	R/W	<Reserved; always write 0x00 (or 5’b00000) to these bits>	0x00
rfmac_synth_tx	2	R/W	Synth TX: 1 for TX mode; 0 for RX mode (manual control via system bus to the digital synthesizer controller)	0x1
–	1:0	R/W	<Reserved; always write 0x0 (or 2’b00) to these bits>	0x0

9.4.11 Clock Enable Tests

Short Name: CLK_TEST
Address: 0x131
Always-On: No

Table 61 • **CLK_TEST**

Field Name	Bit	Access	Description	Default
–	7:4	R/W	<Reserved; always write 0x00 (or 4’b0000) to these bits>	0x00
sys_clk_en_test	3	R/W	Connect system clock output (***cg_sys_clk***) from the clock generator to **GP1**	0x0
–	1:0	R/W	<Reserved; always write 0x0 (or 2’b00) to these bits>	0x0

1. The system clock ***cg_sys_clk*** should be 1.2 MHz. The PLL clock should be 300 kHz.

9.4.12 ADC Mode Conversion

Short Name: ADC_CONV_START
Address: 0x13A
Always-On: No

Table 62 • **ADC_CONV_START**

Field Name	Bit	Access	Description	Default
–	7:2	R	<Reserved>	0x00
single_avg_conv	1	R/W/ CoS	Single ADC average conversion enable. When set high, start a single ADC average conversion. The number of ADC samples used for the average is set by $2^{adc\_pow\_n\_conv}$ defined in **ADC_POW_N_CONV**. An ADC conversion for only one sample can be run by setting *adc_pow_n_conv* to 0. Automatically stopped at the end of the conversion and bit *single_avg_conv* is cleared when the first conversion starts.	0x0
cont_avg_conv	0	R/W	Continuous ADC average conversion enable. When set high, start continuous ADC average conversions made of back-to-back single ADC average conversions. Stop when *cont_avg_conv* is manually set to 0.	0x0

1. Using RSSI in CSMA-CA mode has higher priority than ADC modes. ADC continuous average conversions has higher priority than ADC single average conversion if they are both started at the same time.
Bit *single_avg_conv* is cleared when first conversion starts and therefore always returns 0 on a read operation.

9.4.13 Maximum Result in ADC Modes and When Using RSSI in CSMA-CA Modes

Short Name: ADC_MAX
Address: 0x13B
Always-On: No

Table 63 • **ADC_MAX**

Field Name	Bit	Access	Description	Default
–	7	R	<Reserved>	0x0
adc_max	6:0	R	ADC maximum value. In ADC modes, maximum value over the last $2^{adc\_pow\_n\_conv}$ conversions. Using RSSI in CSMA-CA modes, maximum value over the last $2^{rssi\_pow\_n\_conv}$ conversions.	0x00

9.4.14 Average Result in ADC Modes and When Using RSSI in CSMA-CA Modes

Short Name: ADC_AVG
Address: 0x13C
Always-On: No

Table 64 • **ADC_AVG**

Field Name	Bit	Access	Description	Default
–	7	R	<Reserved>	0x0
adc_avg	6:0	R	ADC average value. In ADC modes, average value over the last $2^{adc\_pow\_n\_conv}$ conversions. Using RSSI in CSMA-CA modes, average value over the last $2^{rssi\_pow\_n\_conv}$ conversions.	0x00

9.4.15 Trimmed RSSI Average at Completion of DC Restore Process in RSSI Detect Modes with CSMA Not Active

Short Name: ADC_AVG_TRMD_PKT_RSSI
Address: 0x13F
Always-On: No

Table 65 • **ADC_AVG_TRMD_PKT_RSSI**

Field Name	Bit	Access	Description	Default
–	7	R	<Reserved>	0x0
adc_avg_trmd_pkt_rssi	6:0	R	ADC trimmed RSSI average value at the completion of the DC restore. In RSSI detect mode with CSMA not active, value of RSSI captured at the completion of the DC restore process. Not used in ADC modes. For this to have any effect, the RSSI threshold register **ADC_RSSI_THRESH** must be set to a value greater than 0.	0x00

9.4.16 IRQ Status

Short Name: IRQ0, IRQ1,and IRQ2
Addresses: 0x143, 0x144, and 0x145 (respectively)
Always-On: No

Table 66 • **IRQ0**

Field Name	Bit	Access	Description	Default
–	7:3	R	<Reserved>	0x0
csma_done_irq	2	CoR	CSMA done IRQ status	0x0
rx_ack_done_irq	1	CoR	RX acknowledgment process done IRQ status	0x0
rx_pkt_done_irq	0	CoR	RX nonacknowledgment packet process done IRQ status	0x0

Table 67 • **IRQ1**

Field Name	Bit	Access	Description	Default
synth_rdy_irq	7	CoR	Synthesizer ready IRQ status	0x0
trim_done_irq	6	CoR	Trim done interrupt status. All the trim routines (except the continuous frequency trim) have determinate duration and should result in this interrupt when all activated rim operations are complete. Make sure all settings necessary for the trim are correct before trying again. See Section 7 Calibrations, page 40, for details of setup.	0x0
trim_fail_irq	5	CoR	Trimming and tune process failed interrupt status. This interrupt occurs whenever a trim ends up at one of its limits. It might not indicate a failure but generally does mean a failure of the trim for some reason.	0x0
adc_avg_done_irq	4	CoR	ADC average done interrupt status. In single ADC average conversion mode and continuous ADC average conversion mode, *adc_avg_done_irq* shall be generated every $2^{adc\_pow\_n\_conv}$ conversions. Using RSSI in CSMA-CA modes, *adc_avg_done_irq* is not generated.	0x0
rssi_nosig_irq	3	CoR	RSSI no signal interrupt status. In single ADC average conversion mode and continuous ADC average conversion mode, *rssi_nosig_irq* is generated every $2^{adc\_pow\_n\_conv}$ conversions if RSSI input rx_rssi is selected and the RSSI average value is less than the selected threshold. When CSMA operation is active, the threshold is ADC_CSMA_THRESH. In other cases, the threshold is **ADC_RSSI_THRESH**.	0x0

Table 67 • **IRQ1 *(continued)***

Field Name	Bit	Access	Description	Default
sync_detect_irq	2	CoR	Frame synchronization detect IRQ status	0x0
pream_det_irq	1	CoR	Preamble detect IRQ status. Goes high during receive when the preamble is detected.	0x0
rssi_high_irq	0	CoR	RSSI high interrupt status. In single ADC average conversion mode and continuous ADC average conversion mode, *rssi_high_irq* is generated every $2^{adc\_pow\_n\_conv}$ conversions if RSSI input rx_rssi is selected and the RSSI average value is greater than or equal to the selected threshold. When CSMA operation is active, the threshold is ADC_CSMA_THRESH. In other cases, the threshold is **ADC_RSSI_THRESH**.	0x0

Table 68 • **IRQ2**

Field Name	Bit	Access	Description	Default
tx_ack_done_irq	7	CoR	TX acknowledgment process done	0x0
tx_pkt_done_irq	6	CoR	TX nonacknowledgment packet process done IRQ status	0x0
pll_lock_err_irq	5	CoR	PLL lock error IRQ status	0x0
rx_hdr_rdy_irq	4	CoR	RX header / buffer ready IRQ status	0x0
rx_frm_pend_irq	3	CoR	Frame pending IRQ status. The is asserted for FP set in a packet and a subsequent packet is not received, a failed data request after retries, or a failed RX after TN and FP are both set on an acknowledgment packet. This IRQ on a node implies that a data request packet should be sent.	0x0
rx_pkt_rdy_irq	2	CoR	RX packet ready in RX buffer IRQ status	0x0
cmd_fail_irq	1	CoR	SPI MAC command fail IRQ status. Set on any packet error of any type, including packet timeout, but not necessarily frame sync timeout if it does not result in a packet timeout.	0x0
cmd_done_irq	0	CoR	SPI MAC command complete IRQ status. The interrupt *cmd_done_irq* is the last interrupt to be asserted in a packet transaction, so it may be best to keep other interrupts on the node disabled and used as status that can be checked when *cmd_done_irq* is asserted.	0x0

9.4.17 Trim Command

Short Name: TRIM_CMD
Address: 0x147
Always-On: No

Table 69 • **TRIM_CMD**

Field Name	Bit	Access	Description	Default
–	7:4	R	<Reserved>	0x0
trim_cmd	3:0	R/W	A write to *trim_cmd* starts the trim indicated below: • 0x1: current reference trim • 0x2: VCO full tune all modes • 0x3: VCO fine tune SAR in RX mode • 0x4: VCO fine tune SAR in TX PA off mode • 0x5: VCO fine tune periodic one-bit in TX PA on mode • 0x6: VCO amplitude trim • 0x7: frequency modulator deviation trim • 0x8: blocker peak detector offset trim • 0x9: LNA load tune • 0xA: antenna tune • 0xB: transconductor (IF filter, FM detector, and Gaussian filter) trim • 0xB: RC oscillator tune • Other: no function	0x0

9.4.18 LNA Trim Peak ADC Value

Short Name: LNA_PEAK_ADC
Address: 0x14A
Always-On: No

Table 70 • **LNA_PEAK_ADC**

Field Name	Bit	Access	Description	Default
–	7:5	R	<Reserved>	0x0
lna_peak_adc	4:0	R	Peak (highest) value from the ADC during the latest LNA tuning	0x00

9.4.19 Antenna Trim Peak ADC Value

Short Name: ANT_PEAK_ADC
Address: 0x14B
Always-On: No

Table 71 • **ANT_PEAK_ADC**

Field Name	Bit	Access	Description	Default
–	7:5	R	<Reserved>	0x0
ant_peak_adc	4:0	R	Peak (highest) value from the ADC during the latest antenna tuning **Note:** It is recommended that the default value (or reset value) be used. Always write 0x00 (or 5’b00000) to these bits.	0x00

9.5 Always-On System Bus Control and Status Registers

Addresses 0x200 to 0x29F comprise the always-on system bus control and status registers, which use long addressing mode.

These registers contain information that is essential for the wake-up process and are therefore powered directly from VSUP (that is, these registers are always-on registers).

9.5.1 VCO Frequency Tune Value for RX Mode

Short Name: VCO_FRQ_RX_TRIM (comprising VCO_FRQ_RX_TRIM_L and VCO_FRQ_RX_TRIM_H)
Addresses: Two-byte little-endian starting at 0x201 (comprising 0x201 and 0x202, respectively)
Always-On: Yes

Table 72 • **VCO_FRQ_RX_TRIM**

Field Name	Bit	Access	Description	Default
–	15:11	R	<Reserved>	0x00
vco_frq_rx	10:0	R/W	VCO frequency tune value for the RX mode **Note:** *vco_frq_trim* is directly driven by value *vco_frq_rx* when internal signals {***pa_tx_en***,*rfmac_synth_tx*}=2’b00 or 2’b10. So, in order to prevent intermediary 11-bit values from controlling the VCO input, *vco_frq_trim* drives a new value of *vco_frq_rx* only when the MSB is written (that is, **VCO_FRQ_RX_TRIM_H** at address 0x202). This means that the LSB (in **VCO_FRQ_RX_TRIM_L** at address 0x201) must be written first, then the MSB to update the 11-bit VCO input.	0x088

9.5.2 VCO Frequency Tune Value for TX mode (with modulation off) with Power Amplifier Off (only bias on)

Short Name: VCO_FRQ_TXPAOFF_TRIM (comprising VCO_FRQ_TXPAOFF_TRIM_L and VCO_FRQ_TXPAOFF_TRIM_H)
Addresses: Two-byte little-endian starting at 0x203 (comprising 0x203 and 0x204, respectively)
Always-On: Yes

Table 73 • **VCO_FRQ_TXPAOFF_TRIM**

Field Name	Bit	Access	Description	Default
–	15:11	R	<Reserved>	0x00
vco_frq_txpaoff	10:0	R/W	VCO frequency tune value for the TX PA off mode. It is with modulation off with the power amplifier off (only bias on). **Note:** *vco_frq_trim* is directly driven by value *vco_frq_txpaoff* when internal signals {***pa_tx_en***,*rfmac_synth_tx*}=2’b01. So, in order to prevent intermediary 11-bit values from controlling the VCO input, *vco_frq_trim* drives a new value of *vco_frq_txpaoff* only when the MSB is written (that is, VCO_FRQ_TXPAOFF_TRIM_H at address 0x204). This means that the LSB (in VCO_FRQ_TXPAOFF_TRIM_L at address 0x203) must be written first, then the MSB to update the 11-bit VCO input.	0x097

9.5.3 VCO Frequency Tune Value for TX mode (with modulation off) with Power Amplifier On

Short Name: VCO_FRQ_TXPAON_TRIM (comprising VCO_FRQ_TXPAON_TRIM_L and VCO_FRQ_TXPAON_TRIM_H)
Addresses: Two-byte little-endian starting at 0x205 (comprising 0x205 and 0x206, respectively)
Always-On: Yes

Table 74 • **VCO_FRQ_TXPAON_TRIM**

Field Name	Bit	Access	Description	Default
–	15:11	R	<Reserved>	0x00
vco_frq_txpaon	10:0	R/W	VCO frequency tune value for the TX PA on mode. It is with modulation off with the power amplifier on. **Note:** *vco_frq_trim* is directly driven by value *vco_frq_txpaon* when internal signals {***pa_tx_en***,*rfmac_synth_tx*}=2’b11. So, in order to prevent intermediary 11-bit values from controlling the VCO input, *vco_frq_trim* drives a new value of *vco_frq_txpaon* only when the MSB is written (that is, VCO_FRQ_TXPAON_TRIM_H at address 0x208). This means that the LSB (in VCO_FRQ_TXPAON_TRIM_L at address 0x205) must be written first, then the MSB to update the 11-bit VCO input.	0x498

9.5.4 VCO Frequency Tune Value at Output of Multiplexer

Short Name: VCO_FRQ_TRIM (comprising VCO_FRQ_TRIM_L and VCO_FRQ_TRIM_H)
Addresses: Two-byte little-endian starting at 0x207 (comprising 0x207 and 0x208, respectively)
Always-On: Yes

Table 75 • **VCO_FRQ_TRIM**

Field Name	Bit	Access	Description	Default
–	15:11	R	<Reserved>	0x00
vco_frq_trim	10:0	R	Frequency tune port at input to VCO (read-only) **Note:** *vco_frq_trim* is directly driven by value *vco_frq_rx* when internal signals {***pa_tx_en***,*rfmac_synth_tx*}=2’b00 or 2’b10, by value *vco_frq_txpaoff* when internal signals {***pa_tx_en***,*rfmac_synth_tx*}=2’b01 and by value *vco_frq_txpaon* when internal signals {***pa_tx_en***,*rfmac_synth_tx*}=2’b11. See programming details in for.*vco_frq_rx*, *vco_frq_txpaoff*, and *vco_frq_txpaon*.	0x_FF

9.5.5 VCO Frequency Band Trim Value

Short Name: VCO_FRQ_BAND_TRIM
Address: 0x209
Always-On: Yes

Table 76 • **VCO_FRQ_BAND_TRIM**

Field Name	Bit	Access	Description	Default
–	7:4	R	<Reserved>	0x0
vco_frq_band_trim	3:0	R/W	VCO frequency band trim value	0x7

9.5.6 Modulator DAC Trim Value

Short Name: MOD_DAC_TRIM
Address: 0x20A
Always-On: Yes

Table 77 • **MOD_DAC_TRIM**

Field Name	Bit	Access	Description	Default
–	7:5	R	<Reserved>	0x0
mod_dac_trim	4:0	R/W	VCO FSK deviation trim value	0x00

9.5.7 Frequency Deviation Trim Target

Short Name: FSK_DEV_TRIM_TARGET
Address: 0x20C
Always-On: Yes

Table 78 • **FSK_DEV_TRIM_TARGET**

Field Name	Bit	Access	Description	Default
fsk_dev_trim_target	7:0	R/W	FSK deviation trim target (default target is mod index 0.5)	0x66

1. For recommended initial register setting, see "Recommended Value" column of Table 35, page 53. The recommended value for this register assumes a 200-bit/s bit rate and a 24-MHz crystal.

9.5.8 RC Oscillator Frequency Trim Value

Short Name: RCOSC_FREQ_TRIM
Address: 0x20D
Always-On: Yes

Table 79 • **RCOSC_FREQ_TRIM**

Field Name	Bit	Access	Description	Default
–	7:6	R	<Reserved>	0x0
rcosc_freq_trim	5:0	R/W	RC oscillator frequency trim value	0x00

9.5.9 LNA Frequency Trim

Short Name: LNA_FRQ_TRIM
Address: 0x20E
Always-On: Yes

Table 80 • **LNA_FRQ_TRIM**

Field Name	Bit	Access	Description	Default
–	7:5	R	<Reserved>	0x0
lna_frq_trim	4:0	R/W	Binary trimming code for tuning LNA resonance frequency	0x16

1. The LNA load is an inductor that should be tuned to obtain the highest gain from the LNA. Higher trim values represent higher capacitance and therefore lower resonant frequency. Use the trim and tune block to automatically trim. See Section 7.5.7 LNA Load Tuning, page 49.

9.5.10 Manual LNA Trim Value

Short Name: LNA_BIAS_TRIM
Address: 0x20F
Always-On: Yes

Table 81 • **LNA_BIAS_TRIM**

Field Name	Bit	Access	Description	Default
–	7:6	R	<Reserved>	0x0
lna_bias_trim	5:2	R/W	Binary trimming code for LNA bias current. Higher gain LNA modes (decimal 4 to 15) should be used only when lna_gain_ctrl is set to highest gain. **Note:** For normal LNA gain, it is recommended that the default value of 0x1 (or 4’b001) be used. For high LNA gain, it is recommended that 0xA (or 4’b1010) be written to these bits upon initialization and after every chip reset.	0x1
mix_bias_trim	1:0	R/W	Binary trimming code for mixer bias current	0x1

1. LNA and mixer biases can be set higher to get more gain on the receiver front-end but more current is used. The default is recommended when trying to stay below 2mA for chip current consumption.

9.5.11 IREF Resistor Trim Value

Short Name: IREF_TRIM
Address: 0x210
Always-On: Yes

Table 82 • **IREF_TRIM**

Field Name	Bit	Access	Description	Default
–	7:5	R	<Reserved>	0x0
iref_trim	4:0	R/W	Resistor trim bus	0x0F

1. A higher trim code means a higher resistor value (and lower current).

9.5.12 Crystal Oscillator Trim Value

Short Name: XO_TRIM
Address: 0x211
Always-On: Yes

Table 83 • **XO_TRIM**

Field Name	Bit	Access	Description	Default
–	7:6	R	<Reserved>	0x0
xo_trim	5:0	R/W	Crystal oscillator trim code	0x26

1. A higher trim code means higher frequency.

9.5.13 Gaussian Filter Trim Value

Short Name: GAUS_TRIM
Address: 0x214
Always-On: Yes

Table 84 • **GAUS_TRIM**

Field Name	Bit	Access	Description	Default
gaus_gm_trim	7:0	R/W	Transconductor tuning voltage	0x62

1. Filter width BT=0.7, so there may be no need to overwrite calibration value with 0xFF.

9.5.14 VCO Amplitude Trim Value

Short Name: VCO_AMP_TRIM
Address: 0x215
Always-On: Yes

Table 85 • **VCO_AMP_TRIM**

Field Name	Bit	Access	Description	Default
–	7:6	R	<Reserved>	0x0
vco_amp_trim	5:0	R/W	Current level control	0x3F

1. Higher trim value means a larger amplitude of oscillations in the VCO. The default is at the highest level to make sure the VCO starts. The trim and tune block should be used to trim the amplitude to 300mV, 350mV, 400mV or 450mV.

9.5.15 Antenna Trim Value

Short Name: ANT_TRIM
Address: 0x216
Always-On: Yes

Table 86 • **ANT_TRIM**

Field Name	Bit	Access	Description	Default
–	7:5	R	<Reserved>	0x0
ant_trim	4:0	R/W	Binary code for antenna resonance frequency tuning	0x16

1. This trim is for the peak detector DC offset. Higher values cause positive offset. The trim and tune block should be used to automatically trim this block. See Section 7.5.6 Antenna Tuning, page 48.

9.5.16 RX Peak Detector Trim Value

Short Name: RX_PKDET_TRIM
Address: 0x217
Always-On: Yes

Table 87 • **RX_PKDET_TRIM**

Field Name	Bit	Access	Description	Default
–	7:5	R	<Reserved>	0x0
rx_pkdet_trim	4:0	R/W	Binary trimming code for peak detector scale (rng=0)	01001

9.5.17 Initial DC Correct/AFC Value

Short Name: DC_CNTR_TRIM
Address: 0x218
Always-On: Yes

Table 88 • **DC_CNTR_TRIM**

Field Name	Bit	Access	Description	Default
dc_cntr_trim	7:0	R/W	Initial DC correct / AFC center (trimmed) value	0x62

1. This value, [written by the FM detector trim function], represents the trimming needed to take out process variations but not offsets due to transmitter carrier frequency offset.

9.5.18 IRQ Enables

Short Name: IRQ_EN0, IRQ_EN1, and IRQ_EN2
Addresses: 0x219, 0x21A, and 0x21B (respectively)
Always-On: Yes

Table 89 • **IRQ_EN0**

Field Name	Bit	Access	Description	Default
–	7:4	R/W	<Reserved>	0x00
spi_rdy_irq_en	3	R/W	SPI ready IRQ enable	0x1
csma_done_irq_en	2	R/W	CSMA done IRQ enable	0x0
rx_ack_done_irq_en	1	R/W	RX acknowledgment process done IRQ enable	0x0
rx_pkt_done_irq_en	0	R/W	RX packet process done IRQ enable	0x0

Table 90 • **IRQ_EN1**

Field Name	Bit	Access	Description	Default
synth_rdy_irq_en	7	R/W	Synthesizer ready IRQ enable	0x0
trim_done_irq_en	6	R/W	Trim done interrupt enable	0x0
trim_fail_irq_en	5	R/W	Trimming and tune process failed interrupt enable	0x0
adc_avg_done_irq_en	4	R/W	ADC average done interrupt enable	0x0
rssi_nosig_irq_en	3	R/W	RSSI no signal interrupt enable	0x0

Table 90 • **IRQ_EN1** ***(continued)***

Field Name	Bit	Access	Description	Default
sync_detect_irq_en	2	R/W	Frame synchronization detect interrupt enable	0x0
pream_det_irq_en	1	R/W	Preamble detect IRQ enable	0x0
rssi_high_irq_en	0	R/W	RSSI high interrupt enable	0x0

Table 91 • **IRQ_EN2**

Field Name	Bit	Access	Description	Default
tx_ack_done_irq_en	7	R/W	TX acknowledgment done IRQ enable	0x0
tx_pkt_done_irq_en	6	R/W	TX nonacknowledgment packet done IRQ enable	0x0
pll_lock_err_irq_en	5	R/W	PLL lock error IRQ enable	0x0
rx_hdr_rdy_irq_en	4	R/W	RX header / buffer ready IRQ enable	0x0
rx_frm_pend_irq_en	3	R/W	RX frame pending IRQ enable	0x0
rx_pkt_rdy_irq_en	2	R/W	RX packet ready in buffer IRQ enable	0x0
cmd_fail_irq_en	1	R/W	SPI command fail IRQ enable	0x0
cmd_done_irq_en	0	R/W	SPI command done IRQ enable	0x0

9.5.19 Pad Enable

Short Name: PAD_EN0
Addresses: 0x21C
Always-On: Yes

Table 92 • **PAD_EN0**

Field Name	Bit	Access	Description	Default
gpio_oen	7:4	R/W	GPIO pin output enable. Tristate control of the **GP3..0** pins. For each bit, setting to 1 means signal is driven out on GPIO and setting to 0 means pin is tristated (high impedance). • If **PAD_EN0[7]** is 1 then pin **GP3** is enabled • If **PAD_EN0[6]** is 1 then pin **GP2** is enabled • If **PAD_EN0[5]** is 1 then pin **GP1** is enabled. Write 0 to this bit if **GP1** is used for TX data input (that is, if **PAD_EN0[1]** is 1) • If **PAD_EN0[4]** is 1 then pin **GP0** is enabled	0x0
gpio_ien	3:0	R/W	GPIO pin input enable. • 0000: Input is disabled for all **GP3..0** pins, which are logically gated off to prevent driving signals into the internal logic. • 0010: Used in raw bit mode to enable TX data input on **GP1**. • Others: not supported	0x0

9.5.20 TX Control

Short Name: TX_CTRL0, TX_CTRL1, and TX_CTRL2
Addresses: 0x21E, 0x21F, and 0x220 (respectively)
Always-On: Yes

Table 93 • **TX_CTRL0**

Field Name	Bit	Access	Description	Default
pkt_retry_max	7:4	R/W	Maximum number of packet retries per transaction sequence. Valid for all packet modes.	0x1
hme_enc_en	3	R/W	Enable hamming encoding on TX. Valid for all packet modes.	0x1
tx_mode	2:0	R/W	Transmit packet mode: • 000: TX in raw bit mode, no frame sync, no length • 001: TX in raw packet mode, fixed length or no length • 010: TX in user packet mode, length in PHY header • 011: Z-Star packet mode, normal frame format • 1XX: Not supported (reserved)	0x3

Table 94 • **TX_CTRL1**

Field Name	Bit	Access	Description	Default
–	7	R	<Reserved>	0x0
tx_reserved_1	6	R/W	Reserved TX MAC header bit, transmitted in TX MAC header in Z-Star packet mode. Always write 0 to this bit.	0x0
tx_pkt_frm_pend	5	R/W	For Z-Star packet mode, set TX frame pending (FP) bit transmitted in MAC header for nonacknowledgment packets. Used only for TX frames, nonacknowledgment, non-data-request. TX1 and TX2	0x0
tx_ack_frm_pend	4	R/W	For Z-Star packet mode, set TX frame pending (FP) bit for acknowledgment packets. Used only for acknowledgment frame of RX command, TX_ACK1	0x0
tx_only	3	R/W	Transmit once only, and then exit. Valid for all packet modes. **Note:** Typically set to 1 on hub.	0x0
tx_null_frm	2	R/W	Send null frame in response to data request. Valid only for TX frame of RX command, TX2. Setting this bit automatically sets the AR bit in the TX MAC header to 0 in Z-Star packet mode.	0x0
tx_tn	1	R/W	For Z-Star packet mode, set TX transmit now (TN) bit in TX MAC header, TX packet immediately after current TX. Used only for acknowledgment frame of RX command, TX_ACK1. This is the same bit location as the acknowledgment request (AR) bit.	0x0
tx_ack_req	0	R/W	For Z-Star packet mode, set TX acknowledgment request (AR) bit in TX MAC header. Not used in acknowledgment frame, data request frame, and null data frame. Used in TX1 and TX2	0x1

1. Note that TX1, TX2, TX_ACK1, and TX_ACK2 correspond to the state machine diagram shown in Figure 14, page 23.

Table 95 • **TX_CTRL2**

Field Name	Bit	Access	Description	Default
data_req_retry_en	7	R/W	For Z-Star packet mode, enable retries on data request when RX packet fails in RX2	0x1
–	6	R/W	<Internal bit; always write 0 to this bit>	0x0
tx_auto_hdr	5	R/W	For Z-Star packet mode, automatic PHY/MAC header generation. By default, auto header is always used in Z-Star for nonacknowledgment, nonnull, and non-data-request packets. For acknowledgments, null and data request packets, auto-header generation is always used. The only exception is if these packet types are sent in TX1 with a TX command and *tx_auto_hdr* = 0. For data request packets and acknowledgment packet in TX_ACK2: payload=0, AR=0, FP=0, FSN=last RX FSN+1 for data request or last RX FSN for acknowledgment, SFRM=0, fcs_len=2. For acknowledgment packets in TX_ACK1: payload=0, FP=tx_ack_frm_pend, AR=tx_tn, FSN=RX FSN, SFRM=0, fcs_len=2. For null packets, payload=0, FP=0, AR=0, FSN=0, FRM=DATA, SFRM=0, fcs_len=2. For nonacknowledgment, nonnull, non-data-request packets: • If *tx_auto_hdr* = 1, all info comes from always-on registers • If *tx_auto_hdr* = 0, all info comes from TX buffer, except length This bit is valid only for Z-Star packet mode as described above and for user packet mode with the following settings. For user packet mode with an eight-bit frame length field and one byte of PHY header and no offset, setting this bit will automatically generate the PHY header on TX.	0x1

[1] Note that TX1, TX2, TX_ACK1, and TX_ACK2 correspond to the state machine diagram shown in Figure 14, page 23.

Table 95 • **TX_CTRL2** ***(continued)***

Field Name	Bit	Access	Description	Default
tx_forever	4	R/W	Restart transmit operation after current transaction	0x0
tx_secure_en	3	R/W	In Z-Star packet mode, set security/encryption bit in TX MAC header	0x0
tx_fcs_len	2:0	R/W	TX FCS length (frame check sequence) CRC byte count in TX packet • 000: No FCS (valid for all packet modes except Z-Star) • 001: One-byte FCS (valid for all packet modes except Z-Star) • 010: Two-byte FCS (valid for all packet modes including Z-Star) • 011: Three-byte FCS (valid for all packet modes except Z-Star) • 100: Four-byte FCS (valid for all packet modes including Z-Star) In Z-Star packet mode, bit [2] is the FCS length bit transmitted in the PHY header. In user and raw packet mode, the *tx_fcs_len* for the transmitting device must be equal to the *rx_fcs_len* on the receiving device.	0x2

1. Note that TX1, TX2, TX_ACK1, and TX_ACK2 correspond to the state machine diagram shown in Figure 14, page 23.

9.5.21 TX Frame-Packet Buffer Length

Short Name: TX_BUF_LEN (comprising TX_BUF_LEN0 and TX_BUF_LEN1)
Addresses: Two-byte little-endian starting at 0x221 (comprising 0x221 and 0x222, respectively)
Always-On: Yes

Table 96 • **TX_BUF_LEN**

Field Name	Bit	Access	Description	Default
–	15:9	R	<Reserved>	0x00
tx_buf_len	8:0	R/W	TX frame-packet buffer length. Number of bytes in transmit buffer for transmission. This value is automatically updated on a write packet command if *tx_auto_len_en* equals 1.	0x00

9.5.22 TX Frame Control for Z-Star Packet Mode

Short Name: TX_FRM_CTRL
Address: 0x223
Always-On: Yes

Table 97 • **TX_FRM_CTRL**

Field Name	Bit	Access	Description	Default
tx_addr_mode	7	R/W	Addressing mode. Controls bit setting in TX MAC header, and address insertion into TX MAC header. • 0: Short addressing mode • 1: Long addressing mode	0x0
tx_reserved_0	6	R/W	Reserved TX MAC header bit, transmitted in MAC header. Always write 0 to this bit.	0x0
tx_frm_stype	5:3	R/W	Frame subtype for TX MAC header; see below for "Frame Type - Frame Subtype" definition in description of field *tx_frm_type*	0x0
tx_frm_type	2:0	R/W	Frame type setting for TX MAC header: **Frame Type - Frame Subtype : Description** • 000 - 000 : Beacon frame • 001 - xxx : Data frame (default value) • 010 - 000 : Acknowledgment frame • 011 - 001 : Association request frame • 011 - 010 : Association response frame • 011 - 011 : Disassociation request frame • 011 - 100 : Data request frame • 011 - 111 : Beacon request frame • 100 - 000 : Channel table request frame • 100 - 001 : Channel table frame • 100 - 010 : Channel change command frame • 100 - 011 : Link quality request frame • 100 - 100 : Link quality data frame	0x1

9.5.23 TX Frame Sequence Number for Z-Star Packet Mode

Short Name: TX_FRM_SEQ_NO
Address: 0x224
Always-On: Yes

Table 98 • **TX_FRM_SEQ_NO**

Field Name	Bit	Access	Description	Default
–	7:4	R	<Reserved>	0x0
tx_frm_seq_no	3:0	R/W	TX frame sequence number setting for TX MAC header for nonacknowledgment and non-data-request frames. This may be autoincremented by the SPI command if not disabled with tx_fsn_incr_dis = 1	0x0

9.5.24 RX Control for Z-Star Packet Mode

Short Name: RX_CTRL0, RX_CTRL1, and RX_CTRL2
Addresses: 0x225, 0x226, and 0x227 (respectively)
Always-On: Yes

Table 99 • **RX_CTRL0**

Field Name	Bit	Access	Description	Default
match_ntwrk	7	R/W	Must match network ID on long addressing	0x0
match_lsrc	6	R/W	Must match source address on long addressing	0x0
shrt_brdcst_en	5	R/W	Enable receiving short-address broadcast.Unmasked portion of network ID must match.	0x0
lng_brdcst_en	4	R/W	Enable receiving long-address broadcast.Network ID is not used for matching. Set to 1 on the hub.	0x0
hmd_dec_en	3	R/W	Enable hamming decoding on receiver.	0x1
rx_mode	2:0	R/W	Receive packet mode: • 000: RX in raw bit mode, no frame sync, no length • 001: RX in raw packet mode, fixed length or no length • 010: RX in user packet mode, length in PHY header • 011: Z-Star packet mode, normal frame format • 1XX: Not supported (reserved)	0x3

Table 100 • **RX_CTRL1**

Field Name	Bit	Access	Description	Default
–	7	R/W	<Internal bit; always write 0 to this bit>	0x0
rx_only	6	R/W	Receive once only, then exit.	0x0
dup_frm_irq_en	5	R/W	Enable rx_pkt_rdy interrupt on duplicate frame. Otherwise, duplicate packets do not generate the rx_pkt_rdy IRQ. **Note:** Set to 1 on the hub. **Note:** This is not an interrupt enable for rx_pkt_rdy IRQ. It only enables rx_pkt_rdy IRQ to be generated, if enabled, on a duplicate packet.	0x0

Table 100 • **RX_CTRL1** ***(continued)***

Field Name	Bit	Access	Description	Default
auto_fp_en	4	R/W	Enable auto frame pending response. If TN and FP = 1 in RX_ACK1, then this enables the device of receive the packet (RX2). Otherwise, it exits.	0x1
rx_forever	3	R/W	Receive continuously. Upon exit, return to receive RX1. This setting disables rx_timeout in RX1. Set to 1 on the hub.	0x0
rx_fcs_len	2:0	R/W	RX FCS (frame check sequence) byte count in PHY_PKT mode: • 000: no FCS (not valid for Z-Star) • 001: 1 byte FCS (not valid for Z-Star) • 010: 2 byte FCS (valid for Z-Star) • 011: 3 byte FCS (not valid for Z-Star) • 100: 4 bytes FCS (valid for Z-Star) In user and raw packet mode, the *tx_fcs_len* for the transmitting device must be equal to the *rx_fcs_len* on the receiving device.	0x2

Table 101 • **RX_CTRL2**

Field Name	Bit	Access	Description	Default
pkt_cntr_en	7	R/W	Enable packet counters to be incremented in the always-on section. This includes all the packet statistic counters.	0x1
rx_hdr_rdy_en	6	R/W	Buffer threshold pulse enable, used to generate *rx_hdr_rdy_irq*. Allows generating an interrupt when the RX buffer reaches the specified level of fullness, as defined by *rx_hdr_thresh*. To generate the IRQ, the rx_hdr_rdy_irq_en bit must also be set.	0x0
rx_ignore_addr	5	R/W	Ignore source, destination and network address for packet reception. If set, the address match is not required for good packet detection.	0x0
phdr_in_crc	4	R/W	Include PHY header in user packet CRC calculation. Otherwise, the CRC calculation starts after the PHY header.	0x0
exit_data_req	3	R/W	Exit RX on data request. If a data request packet is received in RX1, this bit forces an exit. Otherwise, it continues to TX2.	0x0

Table 101 • **RX_CTRL2 *(continued)***

Field Name	Bit	Access	Description	Default
wr_rx_payld_only	2	R/W	Write payload only to RX buffer. Setting this bit to 1 omits the PHY header, MAC header, and CRC remainder from being written to the RX buffer. If this bit is cleared then the entire packet, minus preamble and frame sync, is written to the RX buffer. Set to 0 on hub if it is better for the hub to write the header into the TX buffer.	0x1
rx_phdr_len	1	R/W	PHY header length in non-Z-Star packet mode for RX packets: • 0: One byte • 1: Two bytes	0x0
rx_pkt_len_msb	0	R/W	Bit ordering of packet length field in RX PHY header, Set to 1 for MSB first	0x0

9.5.25 Address Mask

Short Name: ADDR_MASK (comprising ADDR_MASK0 and ADDR_MASK1)
Addresses: Two-byte little-endian starting with 0x229 (comprising 0x229 and 0x22A, respectively)
Always-On: Yes

Table 102 • **ADDR_MASK**

Field Name	Bit	Access	Description	Default
addr_mask	15:0	R/W	Address mask. Mask for received source ID in MAC header, if 1, bit used, if 0, bit ignored for address match and qualification. **Note:** Set to 0xFF00 on hub.	0xFFFF

9.5.26 RX Buffer Length

Short Name: RX_FRM_LEN (comprising RX_FRM_LEN0 and RX_FRM_LEN1)
Addresses: Two-byte little-endian starting with 0x22B (comprising 0x22B and 0x22C, respectively)
Always-On: Yes

Table 103 • **RX_FRM_LEN**

Field Name	Bit	Access	Description	Default
–	15:9	R	<Reserved>	0x00
rx_frm_len	8:0	R/W	RX frame length. Includes headers and CRC in length. This is used to terminate packets in raw packet mode, and can be dynamically updated from the SPI during packet reception to set the length of the packet.	0x00

9.5.27 RX Buffer Write Threshold

Short Name: RX_HDR_THRESH
Address: 0x22D
Always-On: Yes

Table 104 • **RX_HDR_THRESH**

Field Name	Bit	Access	Description	Default
rx_hdr_thresh	7:0	R/W	RX buffer fullness threshold for *rx_hdr_rdy_irq*. Determines the number of bytes written to the RX buffer before the IRQ status is set.	0x00

9.5.28 Network ID

Short Name: NETWORK_ID
Address: 0x22E
Always-On: Yes

Table 105 • **NETWORK_ID**

Field Name	Bit	Access	Description	Default
network_id	7:0	R/W	Network ID, should match for both devices unless masked.	0x00

9.5.29 Device Short ID: Address of This Device

Short Name: DEVICE_SID
Address: 0x22F
Always-On: Yes

Table 106 • **DEVICE_SID**

Field Name	Bit	Access	Description	Default
device_sid	7:0	R/W	Device short ID/address of this device Device ID = source ID on TX side, Device ID = Destination ID on RX side	0x00

9.5.30 Short ID: Address of Other Device

Short Name: OTHER_SID
Address: 0x230
Always-On: Yes

Table 107 • **OTHER_SID**

Field Name	Bit	Access	Description	Default
other_sid	7:0	R/W	Short address – ID of other device Destination ID on TX, Source ID on RX	0x00

9.5.31 Long ID of This Device

Short Name: DEVICE_LID (comprising DEVICE_LID0, DEVICE_LID1, DEVICE_LID2, DEVICE_LID3, DEVICE_LID4, DEVICE_LID5, DEVICE_LID6, and DEVICE_LID7)
Addresses: Eight-byte little-endian starting with 0x231 (comprising 0x231, 0x232, 0x233, 0x234, 0x235, 0x236, 0x237, and 0x238, respectively)
Always-On: Yes

Table 108 • **DEVICE_LID**

Field Name	Bit	Access	Description	Default
device_lid	63:0	R/W	Device long ID	0x00000000 00000000

9.5.32 Long ID of Other Device

Short Name: OTHER_LID (comprising OTHER_LID0, OTHER_LID1, OTHER_LID2, OTHER_LID3, OTHER_LID4, OTHER_LID5, OTHER_LID6, and OTHER_LID7)
Addresses: Eight-byte little-endian starting with 0x239 (comprising 0x239, 0x23A, 0x23B, 0x23C, 0x23D, 0x23E, 0x23F, and 0x240, respectively)
Always-On: Yes

Table 109 • **OTHER_LID**

Field Name	Bit	Access	Description	Default
other_lid	63:0	R/W	Other device long ID	0x00000000 00000000

9.5.33 MAC Controls

Short Name: MAC_CTRL
Address: 0x241
Always-On: Yes

Table 110 • **MAC_CTRL**

Field Name	Bit	Access	Description	Default
–	7:5	R	<Reserved>	0x0
mac_pwrdwn_en	4	R/W	Enable automatic power-down of VDDD at command done, putting the device into the SLEEP state	0x0
dp_pwrdwn_en	3	R/W	Enable data port power-down on sniff auto off when auto_off is 1. **Note:** Set to 0 on hub	0x0
crc_seed	2	R/W	CRC polynomial initial value. For non-Z-Star packet modes only. • 1: all ones • 0: all zeros	0x0
msc_blk_en	1	R/W	MAC enable. When low, acts as synchronous reset to MAC. When this bit is high, the PLL is enabled. This bit must be set (high) before a SPI MAC command is received on the SPI bus.	0x0
hub_node_n	0	R/W	Hub enable. This effects the SPI status byte [3]. • 1: hub • 0: node	0x0

9.5.34 CRC Polynomial

Short Name: CRC_POLY (comprising CRC_POLY0, CRC_POLY1, CRC_POLY2, and CRC_POLY3)
Addresses: Four-byte little-endian starting with 0x242 (comprising 0x242, 0x243, 0x244, and 0x245, respectively)
Always-On: Yes

Table 111 • **CRC_POLY**

Field Name	Bit	Access	Description	Default
crc_poly	31:0	R/W	CRC polynomial. Default value is for 16-bit CRC. Longer packets normally require a longer CRC polynomial.	0x00008F01

9.5.35 SPI Control

Short Name: SPI_CTRL
Address: 0x24A
Always-On: Yes

Table 112 • **SPI_CTRL**

Field Name	Bit	Access	Description	Default
–	7:4	R	<Reserved>	0x0
tx_fsn_incr_dis	3	R/W	Disable automatic TX frame seq no increment on SPI packet write operations. **Note:** Set to 1 on hub with multiple nodes; leave at 0 on hub in point-to-point configurations. Set to 0 on node for Z-Star packet mode; this bit is ignored on the node for other modes.	0x0
tx_auto_len_en	2	R/W	Transmit frame/packet length auto generate enable, 1=auto from SPI, the buffer length is calculated and written to the *tx_buf_len* at the end of the packet write.	0x1
auto_tx_cmd	1	R/W	Set TX command on packet buffer write. When 0, a TX MAC command is not automatically generated to the MAC. In this case, the packet write could be followed by a PKT_TX_CMD if a transmit command is needed. Set to 0 on hub for loading the TX buffer after an ACK with TN/FP equal to 1. Typically, the hub does not initiate packets except beacon.	0x1
tx_cmd_imm	0	R/W	Set TX command to take effect immediately after the first byte. When 0, *spis_tx_cmd* is asserted when SPI_SEL_B goes high at the end of the packet write. This needs to be set to 0 if the packet write cannot be completed before the packet header starts its transmission and the packet length has not been written to *tx_buf_len*. Set to 0 on hub for loading the TX buffer after an ACK with TN/FP equal to 1. Typically, the hub does not initiate packets except beacon.	0x1

9.5.36 CSMA Control

Short Name: CSMA_CTRL
Address: 0x24B
Always-On: Yes

Table 113 • **CSMA_CTRL**

Field Name	Bit	Access	Description	Default
–	7	R	<Reserved>	0x0
csma_mode	6	R/W	Enable CSMA mode on transmit. On TX command or data request command, CSMA is performed prior to packet transmission. Set to 0 on hub if the hub is **not** operating in transaction mode. Typically, the HUB does not initiate packets, except beacon. Typically, the hub does not initiate packets except beacon.	0x1
csma_only	5	R/W	Perform only CSMA on TX command, and then exit	0x0
–	4	R/W	<Internal bit; always write 0 to this bit>	0x0
csma_retry_max	3:0	R/W	CSMA retry count (0 to 15)	0x5

9.5.37 CSMA 100-µs Counts Per Interval

Short Name: CSMA_TIME (comprising CSMA_TIME0 and CSMA_TIME1)
Addresses: Two-byte little-endian starting with 0x24C (comprising 0x24C and 0x24D, respectively)
Always-On: Yes

Table 114 • **CSMA_TIME**

Field Name	Bit	Access	Description	Default
–	15:12	R	<Reserved>	0x0
csma_time	11:0	R/W	100µs counts per CSMA interval; an interval is both the sense dwell time and the multiplier for back-off times	0x00A

9.5.38 CSMA Maximum Random Number of Intervals in Back-Off

Short Name: CSMA_MAX_RAND_BACKOFF (comprising CSMA_MAX_RAND_BACKOFF0 and CSMA_MAX_RAND_BACKOFF1)
Addresses: Two-byte little-endian starting with 0x24E (comprising 0x24E and 0x24F, respectively)
Always-On: Yes

Table 115 • **CSMA_MAX_RAND_BACKOFF**

Field Name	Bit	Access	Description	Default
–	15:12	R	<Reserved>	0x0
csma_max_rand_backoff	11:0	R/W	CSMA maximum random number of intervals in the back-off	0x032

1. An interval is defined by *csma_time*.

9.5.39 CSMA Retry Minimum Back-Off in Intervals

Short Name: CSMA_RETRY_MIN_BACKOFF (comprising CSMA_RETRY_MIN_BACKOFF0 and CSMA_RETRY_MIN_BACKOFF1)
Addresses: Two-byte little-endian starting with 0x250 (comprising 0x250 and 0x251, respectively)
Always-On: Yes

Table 116 • **CSMA_RETRY_MIN_BACKOFF**

Field Name	Bit	Access	Description	Default
–	15:12	R	<Reserved>	0x0
csma_retry_min_backoff	11:0	R/W	CSMA retry minimum back-off in intervals	0x019

1. An interval is defined by *csma_time*.

9.5.40 Synchronization Threshold Minimum Number of Bits That Need to Match the 40-Bit Sync Word

Short Name: SYNC_THRESH
Address: 0x252
Always-On: Yes

Table 117 • **SYNC_THRESH**

Field Name	Bit	Access	Description	Default
–	7:6	R	<Reserved>	0x0
sync_thresh	5:0	R/W	Synchronization threshold: minimum number of bits that need to match the sync word (40 bits). This needs to be reduced for smaller frame sync patterns: • For five-byte sync pattern: 0x23 • For four-byte sync pattern: 0x1C • For three-byte sync pattern: 0x15 • For two-byte sync pattern: 0x0E	0x24

1. For recommended initial register setting, see "Recommended Value" column of Table 92, page 81.

9.5.41 Delay Time Count for ptx_tx_trig

Short Name: TX_TRIG_CNT
Address: 0x254
Always-On: Yes

Table 118 • **TX_TRIG_CNT**

Field Name	Bit	Access	Description	Default
tx_trig_dly_cnt	7:0	R/W	Time delay count for asserting ***ptx_tx_trig***. This count starts at the beginning of the preamble, and counts in resolution of 10 µs. When counter counts down to zero, ***ptx_tx_trig*** is asserted and remains high until the end of the packet. Supports TDMA.	

9.5.42 Selection for GP Input/Output

Short Name: GPIO_0_SEL, GPIO_1_SEL, GPIO_2_SEL, and GPIO_3_SEL
Address: 0x261, 0x262, 0x263, and 0x264 (respectively)
Always-On: Yes

***Table 119* • GPIO_0_SEL**

Field Name	Bit	Access	Description	Default
gpio_0_sel	7:0	R/W	Selection for **GP0**	0x00

***Table 120* • GPIO_1_SEL**

Field Name	Bit	Access	Description	Default
gpio_1_sel	7:0	R/W	Selection for **GP1**	0x00

***Table 121* • GPIO_2_SEL**

Field Name	Bit	Access	Description	Default
gpio_2_sel	7:0	R/W	Selection for **GP2**	0x00

***Table 122* • GPIO_3_SEL**

Field Name	Bit	Access	Description	Default
gpio_3_sel	7:0	R/W	Selection for **GP3**	0x00

9.5.43 Frame Sync Pattern

Short Name: SYNC_PTRN (comprising SYNC_PTRN0, SYNC_PTRN1, SYNC_PTRN2, SYNC_PTRN3, and SYNC_PTRN4)
Addresses: Five-byte little-endian starting at 0x265 (comprising 0x265, 0x266, 0x267, 0x268, and 0x269, respectively)
Always-On: Yes

***Table 123* • SYNC_PTRN**

Field Name	Bit	Access	Description	Default
sync_ptrn	39:0	R/W	Frame sync pattern. **Note:** It is recommended that the default value (or reset value) be used.	0x79EC32 DA13

9.5.44 PLL Start-Up Delay (100-µs count)

Short Name: PLL_START_DLY
Address: 0x26A
Always-On: Yes

Table 124 • **PLL_START_DLY**

Field Name	Bit	Access	Description	Default
pll_start_dly	7:0	R/W	PLL start-up delay (100µs counts). This is the time for the PLL to settle when it is first turned on. (Used in protocols and certain trimming commands.)	0x14

9.5.45 RX Time Limit for Acknowledgment Timeout (100-µs count)

Short Name: ACK_TIME_LIMIT
Address: 0x270
Always-On: Yes

Table 125 • **ACK_TIME_LIMIT**

Field Name	Bit	Access	Description	Default
ack_time_limit	7:0	R/W	RX acknowledgment packet timeout for Z-Star packet mode. The timeout period is from when the receiver is enabled until frame sync is detected. This results in a dropped packet. The timeout is disabled with a value of zero. This timeout is automatically scaled for different data rates: • If *rx_rate* is 2’b00, then 100 µs per bit • If *rx_rate* is 2’b01, then 200 µs per bit • If *rx_rate* is 2’b10, then 400 µs per bit	0x18

1. For recommended initial register setting, see "Recommended Value" column of Table 92, page 81. The timeout is sensitive to the preamble length and the response time of the transmitter, when the transmitter is not in transaction mode. For longer preambles, the value of ack_time_limit should be increased.

9.5.46 RX Time Limit for Packet Timeout (100-µs count)

Short Name: PKT_TIME_LIMIT
Address: 0x271
Always-On: Yes

Table 126 • **PKT_TIME_LIMIT**

Field Name	Bit	Access	Description	Default
pkt_time_limit	7:0	R/W	RX nonacknowledgment packet timeout. The timeout period is from when the receiver is enabled until frame sync is detected. This results in a dropped packet. The timeout is disabled with a value of zero. It is also disabled if *rx_forever* is high. This timeout is automatically scaled for different data rates: • If *rx_rate* is 2’b00, then 100 µs per bit • If *rx_rate* is 2’b01, then 200 µs per bit • If *rx_rate* is 2’b10, then 400 µs per bit	0x24

1. For recommended initial register setting, see "Recommended Value" column of Table 92, page 81. The timeout is sensitive to the preamble length and the response time of the transmitter, when the transmitter is not in transaction mode. For longer preambles, the value of pkt_time_limit should be increased.

9.5.47 RX Frame Sequence Number

Short Name: RX_FRM_SEQ_NO
Address: 0x272
Always-On: Yes

Table 127 • **RX_FRM_SEQ_NO**

Field Name	Bit	Access	Description	Default
–	7:4	R	<Reserved>	0x0
rx_frm_seq_no	3:0	R/W	Latest nonacknowledgment received frame sequence number. This is automatically updated by the receiver FSM. Valid only for Z-Star packet mode.	0x0

9.5.48 TX Non-Acknowledgment Packet Transmitted Count

Short Name: TX_PKT_CNT
Address: 0x273
Always-On: Yes

Table 128 • **TX_PKT_CNT**

Field Name	Bit	Access	Description	Default
tx_pkt_cnt	7:0	R/W	Number of nonacknowledgment packets transmitted. Clear on read, freeze at 0xFF. Valid only for Z-Star packet mode.	0x00

9.5.49 TX Packet Retry Accumulated Count (after ACK failure)

Short Name: TX_PKT_RETRY_CNT
Address: 0x274
Always-On: Yes

Table 129 • **TX_PKT_RETRY_CNT**

Field Name	Bit	Access	Description	Default
tx_pkt_retry_cnt	7:0	R/W	Number of TX packet auto-retries (after acknowledgment failure). Clear on read, freeze at 0xFF.	0x00

9.5.50 TX Packet Drop Accumulated Count (after ACK failure)

Short Name: TX_PKT_DROP_CNT
Address: 0x275
Always-On: Yes

Table 130 • **TX_PKT_DROP_CNT**

Field Name	Bit	Access	Description	Default
tx_pkt_drop_cnt	7:0	R/W	Number of TX packet dropped after max auto-retries (after acknowledgment failure). Clear on read, freeze at 0xFF. Results in *cmd_fail_irq*. This is operational only in Z-Star packet mode.	0x00

9.5.51 RX Non-Acknowledgment Packet Received Count

Short Name: RX_PKT_CNT
Address: 0x276
Always-On: Yes

Table 131 • **RX_PKT_CNT**

Field Name	Bit	Access	Description	Default
rx_pkt_cnt	7:0	R/W	Number of nonacknowledgment packets received. Clear on read, freeze at 0xFF. Wrong frame type and duplicate packets are not included. Corresponded to rx_pkt_rdy_irq count.	0x00

9.5.52 RX Packet Received Count for All Types

Short Name: RX_ALL_PKT_CNT
Address: 0x277
Always-On: Yes

Table 132 • **RX_ALL_PKT_CNT**

Field Name	Bit	Access	Description	Default
rx_all_pkt_cnt	7:0	R/W	Number of packets received of all types without error. Clear on read, freeze at 0xFF. Includes duplicate packets, and type error packets, and acknowledgment packets.	0x00

9.5.53 RX Non-Acknowledgment Packet Drop Accumulated Count

Short Name: RX_PKT_DROP_CNT
Address: 0x278
Always-On: Yes

Table 133 • **RX_PKT_DROP_CNT**

Field Name	Bit	Access	Description	Default
rx_pkt_drop_cnt	7:0	R/W	Number of nonacknowledgment RX packet dropped for packet error or timeout after all auto-retries (after acknowledgment failure). Clear on read, freeze at 0xFF. (Sync error is not included if timeout is disabled.) Results in *cmd_fail_irq*.	0x00

9.5.54 RX Packet Sync Error Accumulated Count

Short Name: RX_SYNC_ERR_CNT
Address: 0x279
Always-On: Yes

Table 134 • **RX_SYNC_ERR_CNT**

Field Name	Bit	Access	Description	Default
rx_sync_err_cnt	7:0	R/W	Number of RX packet frame sync errors. Clear on read, freeze at 0xFF. This is typically frame sync timeout after preamble detect, independent of packet timeout. May be from false preamble detect.	0x00

9.5.55 RX Packet All Error Accumulated Count for All Errors

Short Name: RX_ALL_ERR_CNT
Address: 0x27A
Always-On: Yes

Table 135 • **RX_ALL_ERR_CNT**

Field Name	Bit	Access	Description	Default
rx_all_err_cnt	7:0	R/W	Number of RX packets with errors of all types. Clear on read, freeze at 0xFF. Includes all packet timeout, CRC, and frame type errors. Frame sync errors are only included if they result in packet timeout error. Results in *cmd_fail_irq*.	0x00

9.5.56 CSMA Retry Count

Short Name: CSMA_RETRY_CNT
Address: 0x27B
Always-On: Yes

Table 136 • **CSMA_RETRY_CNT**

Field Name	Bit	Access	Description	Default
csma_retry_cnt	7:0	R/W	Number of CSMA retries. Clear on read, freeze at 0xFF.	0x00

9.5.57 CSMA Fail Count

Short Name: CSMA_FAIL_CNT
Address: 0x27C
Always-On: Yes

Table 137 • **CSMA_FAIL_CNT**

Field Name	Bit	Access	Description	Default
csma_fail_cnt	7:0	R/W	Number of CSMA failures after max retries. Clear on read, freeze at 0xFF. Results in *cmd_fail_irq*.	0x00

9.5.58 PHY RX Mode Select

Short Name: PHY_RX_MODE_SEL
Address: 0x282
Always-On: Yes

Table 138 • **PHY_RX_MODE_SEL**

Field Name	Bit	Access	Description	Default
–	7:4	R/W	<Reserved; always write 0x0 (or 4’b0000) to these bits>	0x4
raw_rx_mode	3	R/W	Operate RX in raw bit mode	0x0
–	2:1	R/W	<Reserved; always write 0x2 (or 2’b10) to these bits>	0x2
pream_det_mode	0	R/W	Enable preamble packet detection: • 0: disabled • 1: enabled. **Note:** Set to 1 on hub.	0x1

1. For recommended initial register setting, see "Recommended Value" column of Table 92, page 81.

9.5.59 Frame Sync Control 2

Short Name: DPORT_CTRL
Address: 0x287
Always-On: Yes

Table 139 • **DPORT_CTRL**

Field Name	Bit	Access	Description	Default
–	7:3	R	<Reserved; always write 0x03 (or 5’b00011) to these bits>	0x03
sync_len	2:0	R/W	Length in bytes of frame sync pattern for both TX and RX. This must be the same for both devices. **Note:** For recommended setting, see "Recommended Value" column of Table 35, page 53. The recommended value for these bits assumes three bytes of frame sync.	0x5

9.5.60 PHY TX Raw Mode Control

Short Name: PHY_TX_RAW_MODE_CTRL
Address: 0x288
Always-On: Yes

Table 140 • **PHY_TX_RAW_MODE_CTRL**

Field Name	Bit	Access	Description	Default
–	7:2	R	<Reserved>	0x00
raw_tx_mode	1	R/W	Operate TX in raw bit mode	0x0
gpio_tx_sel	0	R/W	Select GPIO for input of data to be transmitted. Valid only for raw bit mode (TX). • 0: normal (e.g., TX data from TX buffer) • 1: TX data from **GP1**	0x0

9.5.61 Data Rate Control

Short Name: RATE_CTRL
Address: 0x289
Always-On: Yes

Table 141 • **RATE_CTRL**

Field Name	Bit	Access	Description	Default
tx_follow_rx_fec	7	R/W	Transmit in same FEC mode as last packet received	0x0
adapt_fec_en	6	R/W	Enable for adaptive FEC in receive, based on frame sync polarity, negative sync enables FEC. Must be high to transmit inverted sync if other device is in adaptive FEC mode, which means both devices must have the same setting.	0x0
tx_follow_rx_rate	5	R/W	Transmit at rate of last received packet	0x0

Table 141 • **RATE_CTRL** ***(continued)***

Field Name	Bit	Access	Description	Default
adapt_rate_en	4	R/W	Enable for adaptive rate on receiver	0x0
rx_rate	3:2	R/W	RX rate selection: • 00: 200 kbit/s • 01: 100 kbit/s • 10: 50 kbit/s	0x0
tx_rate	1:0	R/W	TX rate selection: • 00: 200 kbit/s • 01: 100 kbit/s • 10: 50 kbit/s	0x0

9.5.62 M Divide Counter Value

Short Name: SYNTH_M_DIV
Address: 0x28A
Always-On: Yes

Table 142 • **SYNTH_M_DIV**

Field Name	Bit	Access	Description	Default
m_div	7:0	R/W	M div counter value: number of times the prescaler counts to 16 within one phase comparison period	0xB5

1. The M divider should be set greater than or equal to 16. See Section 6.4 Synthesizer Controller and Channel Selection, page 36, for instructions on calculating the A and M values for a given channel frequency.
2. This register should be written first, then SYNTH_A_DIV. At that time, the outputs of both registers are updated.

9.5.63 A Divide Counter Value

Short Name: SYNTH_A_DIV
Address: 0x28B
Always-On: Yes

Table 143 • **SYNTH_A_DIV**

Field Name	Bit	Access	Description	Default
–	7:6	R	<Reserved>	0x0
a_div	5:0	R/W	A div counter value: number of times the prescaler counts to 17 within one phase comparison period.	0x07

1. The LO is automatically offset by 600kHz from the channel in use unless bit 1 of address 0x61 is set to 0. The A divider should be set greater than or equal to 5.

9.5.64 ADC Modes Multiplexer Input Selection

Short Name: ADC_MUX_IN_SEL
Address: 0x28E
Always-On: Yes

Table 144 • **ADC_MUX_IN_SEL**

Field Name	Bit	Access	Description	Default
–	7:3	R	<Reserved>	0x00
adc_mux_in_sel	2:0	R/W	In ADC modes, ADC multiplexer input selection: • 000: none of the inputs is selected (analog multiplexer output floats). • 001: RX mixer output (mix_outp). • 010: blocker peak detector output (pd_out). • 100: RSSI output (rx_rssi). • Others: not recommended. If more than one bit is set, more than one switch is turned on according to the input selection. That is not the normal operation. Using RSSI in CSMA-CA modes, these bits are ignored because the finite state machine automatically selects the RSSI input rx_rssi.	0x0

9.5.65 Number of Conversions Used for Averaging in ADC Modes and When Using RSSI in CSMA-CA Modes

Short Name: ADC_POW_N_CONV
Address: 0x28F
Always-On: Yes

Table 145 • **ADC_POW_N_CONV**

Field Name	Bit	Access	Description	Default
–	7	R	<Reserved>	0x0
rssi_pow_n_conv	6:4	R/W	Number of ADC conversions when using RSSI in CSMA-CA modes. When using RSSI in CSMA-CA modes, determine the number of ADC conversions before updating **ADC_MAX** and **ADC_AVG** and generating interrupts ADC average done. • 000: 1 conversion. • 001: 2 conversions. • 010: 4 conversions. • 011: 8 conversions. • 100: 16 conversions. • 101: 32 conversions. • 110: 64 conversions. • 111: 128 conversions. Not used in ADC modes.	0x3
adc_pow_n_conv	3:0	R/W	Number of ADC conversions in ADC modes. In ADC modes, determine the number of ADC conversions before updating **ADC_MAX** and **ADC_AVG** and generating interrupt ADC average done. • 0000: 1 conversion. • 0001: 2 conversions. • 0010: 4 conversions. • 0011: 8 conversions. • 0100: 16 conversions. • 0101: 32 conversions. • 0110: 64 conversions. • 0111: 128 conversions. • 1000: 256 conversions. • 1001: 512 conversions. • 1010: 1024 conversions. • 1011 to 1111: 2048 conversions. Not used in with RSSI in CSMA-CA modes. **Note:** A single ADC conversion takes 78 + $6 \times 2^{adc\_pow\_n\_conv}$ cycles of the clock ***cg_sys_clk***. For example, with the default value, a single ADC conversion takes ($78+6*2^{0}$) clock cycles, which corresponds to $84 \times 1/1.2\,\mu s = 84 \times 0.833\,\mu s = 70\,\mu s$.	0x0

9.5.66 ADC CSMA Threshold

Short Name: ADC_CSMA_THRESH
Address: 0x294
Always-On: Yes

Table 146 • **ADC_CSMA_THRESH**

Field Name	Bit	Access	Description	Default
–	7	R	<Reserved>	0x0
adc_csma_thresh	6:0	R/W	CSMA threshold. Is used to determine whether a channel is clear for transmission. If RSSI signal is below this threshold, then the channel is clear.	0x20

9.5.67 LNA Gain

Short Name: LNA_GAIN
Address: 0x295
Always-On: Yes

Table 147 • **LNA_GAIN**

Field Name	Bit	Access	Description	Default
–	7:4	R	<Reserved>	0x0
lna_gain	3:0	R/W	In all modes, control LNA gain. The gain word is a thermometer code: • 1111 = highest gain • 0111 • 0011 • 0001 • 0000 = lowest gain	0x7

1. For recommended initial register setting, see "Recommended Value" column of Table 92, page 81.

9.5.68 PA Power Level

Short Name: PA_PWR_LEVEL
Address: 0x29A
Always-On: Yes

Table 148 • **PA_PWR_LEVEL**

Field Name	Bit	Access	Description	Default
pa_pwr_level	7:0	R/W	Power control word *pa_pwr_level*[7]: enable PA soft on-off (default is 1, enabled) *pa_pwr_level[6]*: enable DAC scale down by 1/2 (default is 0, disabled). *pa_pwr_level[5:0]*: PA power level.	0x88

9.5.69 PA Buffer Bias Control

Short Name: VCO_BUF_BIAS
Address: 0x29B
Always-On: Yes

Table 149 • **VCO_BUF_BIAS**

Field Name	Bit	Access	Description	Default
–	7:4	R	<Reserved>	0x0
vco_buf_bias	3:0	R/W	Trim bits to adjust bias current in the buffer amplifier of the VCO	0x3

[1] It is recommended that the default (or reset) value be used.

9.5.70 VCO Control

Short Name: VCO_CTRL
Address: 0x29C
Always-On: Yes

Table 150 • **VCO_CTRL**

Field Name	Bit	Access	Description	Default
–	7	R	<Reserved>	0x0
ch_lo_ctrl	6	R/W	1 = LO above channel, 0 = LO below channel	0x0
vco_low_range	5	R/W	Enable low VCO frequency operation	0x0
–	4:0	R/W	<Reserved; always write 0x08 (or 5’b01000) to these bits>	0x08

10 Errata

The Errata section documents exceptions to the specifications documented for the ZL70550 in the Programmer User’s Guide. Each errata item includes:

- Errata title and ID
- Status
- Hardware / software version(s) to which the errata applies
- Description of problem
- Solution or workaround

10.1 Full VCO Trim Failure

The full VCO trim failure (bug #8250) affects part number ZL70550LDF1, chip revision 0x11.

10.1.1 Description

Statement of the problem:

When performing a full VCO trim, it is possible that one of the VCO trims (**VCO_FRQ_RX_TRIM**, **VCO_FRQ_TXPAON_TRIM**, or **VCO_FRQ_TXPAOFF_TRIM**) falls near the edge of a band. The VCO trim algorithm should recognize this and increment or decrement the band (depending on which end of the band limit was reached) to get a more centered VCO trim value. However, due to an error in the VCO trim algorithm, the VCO band increment or decrement does not occur if the VCO band limit is exceeded.

Explanation of the impact:

The band limits are set to two by default. With a setting of two, the **VCO_FRQ_BAND_TRIM** result should be a value between the range of 3 and 2044. If any of the VCO trim values fall outside of this range, the intention of the VCO trim algorithm was to force an increment or decrement of the **VCO_FRQ_BAND_TRIM** register in order to keep the **VCO_FRQ_BAND_TRIM** result closer to the middle of the VCO tuning range. However, whenever any of the VCO trims exceed two, the application is notified by the assertion of the *trim_fail_irq* interrupt.

10.1.2 Solution or Workaround

The workaround is as follows.

If the *trim_fail_irq* interrupt occurs, perform another full VCO trim.

11 References

Document No.	Document Title
–	FCC Part 15 and the European pr ETS 300-220 regulatory documentation
–	Mil-Std-883 Method 3015
EN301 357 – 1	Electromagnetic compatibility and Radio Spectrum Matters (ERM); Cordless audio devices in the range 25 to 2000 MHz; Consumer radio microphones and in-ear monitoring systems operating in the CEPT harmonized band 863 to 865 MHz; Part 1: Technical characteristics and Test methods
EN301 357 – 2	Electromagnetic compatibility and Radio Spectrum Matters (ERM); Cordless audio devices in the range 25 to 2000 MHz; Consumer radio microphones and in-ear monitoring systems operating in the CEPT harmonized band 863 to 865 MHz; Part 2: Harmonized EN under article 3.2 of the R&TTE Directive
EN301 489 – 1	Electromagnetic compatibility and Radio Spectrum Matters (ERM); Electromagnetic Compatibility (EMC) (EMC) standards for radio equipment and services; Part 1: Common technical requirements
EN301 489 – 9	Electromagnetic Compatibility and Radio Spectrum Matters (ERM); Electromagnetic Compatibility (EMC) (EMC) standards for radio equipment and services; Part 9: Specific conditions for wireless microphones, similar Radio Frequency (RF) audio link equipment, cordless audio and in-ear monitoring devices

12 Glossary

Term	Definition
Ack	Acknowledgment
ADC	Analog to Digital Converter
Addr	Address
ADK	Application development kit
AFC	Automatic frequency control
AR	Acknowledgment request
Cmd	Command
CRC	Cyclic redundancy check
CSMA	Carrier Sense Multiple Access
CSMA-CA	Carrier Sense Multiple Access -- Collision Avoidance
DAC	Digital to analog converter
Dec	Decimal
FCS	Frame check sequence
FEC	Forward error correction
FIR	Finite impulse response
FP	Frame pending
FSN	Frame sequence number
Hex	Hexadecimal
I/O	Input/output
IF	Intermediate frequency
IRQ	Interrupt
ISM	Industrial-Scientific-Medical
LBT	Listen before talk
LDO	Low dropout
LNA	Low-noise amplifier
LO	Local oscillator
LSB	Least significant bit
MAC	Media access controller
MPU	Microprocessor unit
MSB	Most significant bit
PA	Power amplifier
PHY	Physical
Pkt	Packet
PLL	Phase locked loop
RC	Resistor-capacitor
RCO	RC oscillator (150-kHz strobe oscillator)

Term	Definition
Rdy	Ready
RSSI	Received signal strength indicator
RX	Receive
SCLA	Source code license agreement
SPI	Serial peripheral interface
TDMA	Time division multiple access
TN	Transmit now
TX	Transmit
VCO	Voltage controlled oscillator
VDDD	VDD digital
XO	Crystal oscillator
XTAL	Crystal
GPIO	General-purpose I/O
VDDA	VDD analog