Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

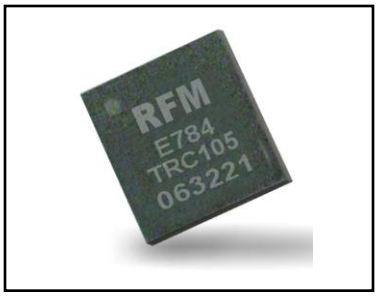

## Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

muRata
INNOVATOR IN ELECTRONICS

# TRC105

## 300-510 MHz RF Transceiver




## Product Overview

TRC105 is a single chip, multi-channel, low power UHF transceiver. It is designed for low cost, high volume, two-way short range wireless applications in the 300 to 510 MHz frequency range. The TRC105 is FCC & ETSI certifiable. All critical RF and base-band functions are integrated in the TRC105, minimizing external component count and simplifying and speeding design-ins. A microcontroller, RF SAW filter, 12.8 MHz crystal and a few passive components are all that is needed to create a complete, robust radio function. The TRC105 incorporates a set of low-power states to reduce current consumption and extend battery life. The small size and low power consumption of the TRC105 make it ideal for a wide variety of short range radio applications. The TRC105 complies with Directive 2002/95/EC (RoHS).

## Key Features

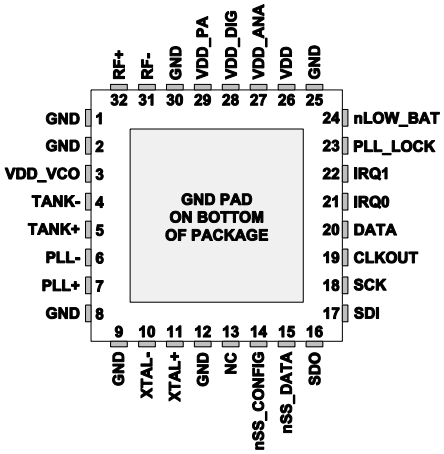
- Modulation: FSK or OOK with frequency hopping spread spectrum capability
- Frequency range: 300 to 510 MHz
- High sensitivity: -112 dBm in circuit
- High data rate: up to 200 kb/s
- Low receiver current: 2.7 mA typical
- Low sleep current: 0.1 µA typical
- Up to +13 dBm in-circuit transmit power
- Operating supply voltage: 2.1 to 3.6 V
- Programmable preamble
- Programmable packet start pattern
- Integrated RF, PLL, IF and base-band circuitry
- Integrated data & clock recovery
- Programmable RF output power
- PLL lock output
- Transmit/receive FIFO size programmable up to 64 bytes
- Continuous, buffered and packet data modes
- Packet address recognition
- Packet handling features:
  - Fixed or variable packet length
  - Packet filtering
  - Packet formatting
- Standard SPI interface
- TTL/CMOS compatible I/O pins
- Programmable clock output frequency
- Low Battery Detection
- Low cost 12.8 MHz crystal reference
- Integrated RSSI
- Integrated crystal oscillator
- Host processor interrupt pins
- Programmable data rate
- External wake-up event inputs
- Integrated packet CRC error detection
- Integrated DC-balanced data scrambling
- Integrated Manchester encoding/decoding
- Interrupt signal mapping function
- Support for multiple channels
- Four power-saving modes
- Low external component count
- TQFN-32 SMT package
- Standard 13 inch reel, 3K pieces

## Applications

- Active RFID tags
- Automated meter reading
- Home & industrial automation
- Security systems
- Two-way remote keyless entry
- Automobile immobilizers
- Sports performance monitoring
- Wireless toys
- Medical equipment
- Low power two-way telemetry systems
- Wireless mesh sensor networks
- Wireless modules

# Table of Contents

## 1.0 Pin Configuration




## 1.1 Pin Description

| PIN | TYPE | NAME | DESCRIPTION |
|---|---|---|---|
| 1 | - | GND | CONNECT TO GND |
| 2 | - | GND | CONNECT TO GND |
| 3 | O | VDD_VCO | REGULATED SUPPLY FOR VCO |
| 4 | I/O | TANK- | VCO TANK |
| 5 | I/O | TANK+ | VCO TANK |
| 6 | I/O | PLL- | PLL LOOP FILTER OUTPUT |
| 7 | I/O | PLL+ | PLL LOOP FILTER INPUT |
| 8 | - | GND | CONNECT TO GND |
| 9 | - | GND | CONNECT TO GND |
| 10 | I/O | XTAL- | CRYSTAL CONNECTION (OSCILLATOR OUTPUT) |
| 11 | I/O | XTAL+ | CRYSTAL CONNECTION (OSCILLATOR INPUT) |
| 12 | - | GND | CONNECT TO GND |
| 13 | - | NC | NO CONNECTION - FLOAT PIN |
| 14 | I | nSS_CONFIG | SLAVE SELECT FOR SPI CONFIGURATION DATA |
| 15 | I | nSS_DATA | SLAVE SELECT FOR SPI TX/RX DATA |
| 16 | O | SDO | SERIAL DATA OUT |
| 17 | I | SDI | SERIAL DATA IN |
| 18 | I | SCK | SERIAL SPI CLOCK IN |
| 19 | O | CLKOUT | BUFFERED CLOCK OUTPUT |
| 20 | I/O | DATA | TRANSMIT/RECEIVE DATA |
| 21 | O | IRQ0 | INTERRUPT OUTPUT |
| 22 | O | IRQ1/DCLK | INTERRUPT OUTPUT/RECOVERED DATA CLOCK (CONT MODE) |
| 23 | O | PLL_LOCK | PLL LOCKED INDICATOR |
| 24 | O | nLOW_BATT | LOW BATTERY DETECT (LEAVE UNCONNECTED IF NOT USED) |
| 25 | - | GND | CONNECT TO GND |
| 26 | I | VDD | MAIN 3.3 V SUPPLY VOLTAGE |
| 27 | O | VDD_ANA | REGULATED SUPPLY FOR ANALOG CIRCUITRY |
| 28 | O | VDD_DIG | REGULATED SUPPLY FOR DIGITAL CIRCUITRY |
| 29 | O | VDD_PA | REGULATED SUPPLY FOR RF POWER AMP |
| 30 | - | GND | CONNECT TO GND |
| 31 | I/O | RF- | RF I/O |
| 32 | I/O | RF+ | RF I/O |
| PAD | - | GROUND | GROUND PAD ON PKG BOTTOM |

Table 1

## 2.0 Functional Description

The TRC105 is a single-chip transceiver that can operate in the 300-510 MHz frequency range. The TRC105 supports two modulation schemes - FSK and OOK. The TRC105's highly integrated architecture requires a minimum of external components, while maintaining design flexibility. All major RF communication parameters are programmable and most can be dynamically set. The TRC105 is optimized for very low power consumption (2.7 mA typical in receiver mode). It complies with European ETSI, FCC Part 15 and Canadian RSS-210 regulatory standards. Advanced digital features including the TX/RX FIFO and the packet handling data mode significantly reduce the load on the host microcontroller.

**TRC105 Block Diagram**

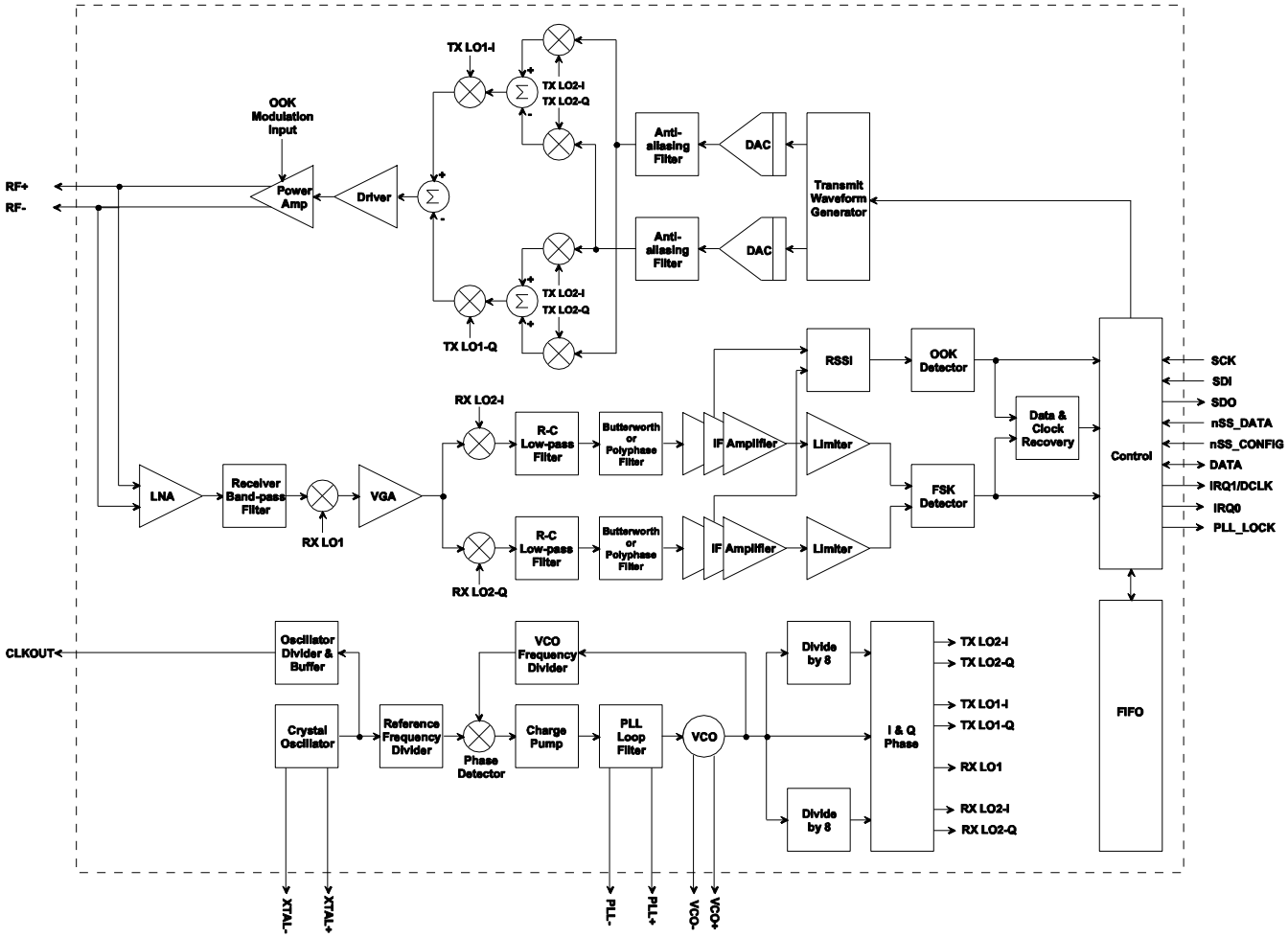



Figure 1

The receiver is based on a superheterodyne architecture. It is composed of the following major blocks:

- An LNA that provides low noise RF gain followed by an RF band-pass filter.
- A first mixer which down-converts the RF signal to an intermediate frequency equal to 1/9th of the carrier frequency.
- A variable gain first IF preamplifier followed by two second mixers which down convert the first IF signal to I and Q signals at a low frequency (zero-IF for FSK, low-IF for OOK).

- A two-stage IF filter followed by an amplifier chain for both the I and Q channels. Limiters at the end of each chain drive the I and Q inputs to the FSK demodulator function. An RSSI signal is also derived from the I and Q IF amplifiers to drive the OOK detector. The second filter stage in each channel can be configured as either a third-order Butterworth low-pass filter for FSK operation or an image reject polyphase band-pass filter for OOK operation.
- An FSK arctangent type demodulator driven from the I and Q limiter outputs, and an OOK demodulator driven by the RSSI signal. Either detector can drive a data and clock recovery function that provides matched filter enhancement of the demodulated data.

The transmitter chain is based on the same double-conversion architecture and uses the same intermediate frequencies as the receiver chain. The main blocks include:

- A digital waveform generator that provides the I and Q base-band signals. This block includes digital-to-analog converters and anti-aliasing low-pass filters.
- A compound image-rejection mixer to up convert the base-band signal to the first IF at 1/9th of the carrier frequency, and a second image-rejection mixer to up-convert the IF signal to the RF frequency
- Transmitter driver and power amplifier stages to drive the antenna port

The frequency synthesizer is based on an integer-N PLL having a typical frequency step size of 12.5 kHz. Two programmable frequency dividers in the feedback loop of the PLL and one programmable divider on the reference oscillator allow the LO frequency to be adjusted. The reference frequency is generated by a crystal oscillator running at 12.8 MHz.

The TRC105 is controlled by a digital block that includes registers to store the configuration settings of the radio. These registers are accessed by a host microcontroller through an SPI style serial interface. The microcontroller's serial connections to the TRC105's SDI, SDO and SCK pins are shown in Figure 2 (component values shown are for 418.00-434.79 MHz operation; see Tables 57 and 58 for other frequency bands). On-chip regulators provide stable supply voltages to sensitive blocks and allow the TRC105 to be used with supply voltages from 2.1 to 3.6 V. Most blocks are supplied with a voltage below 1.6 V.

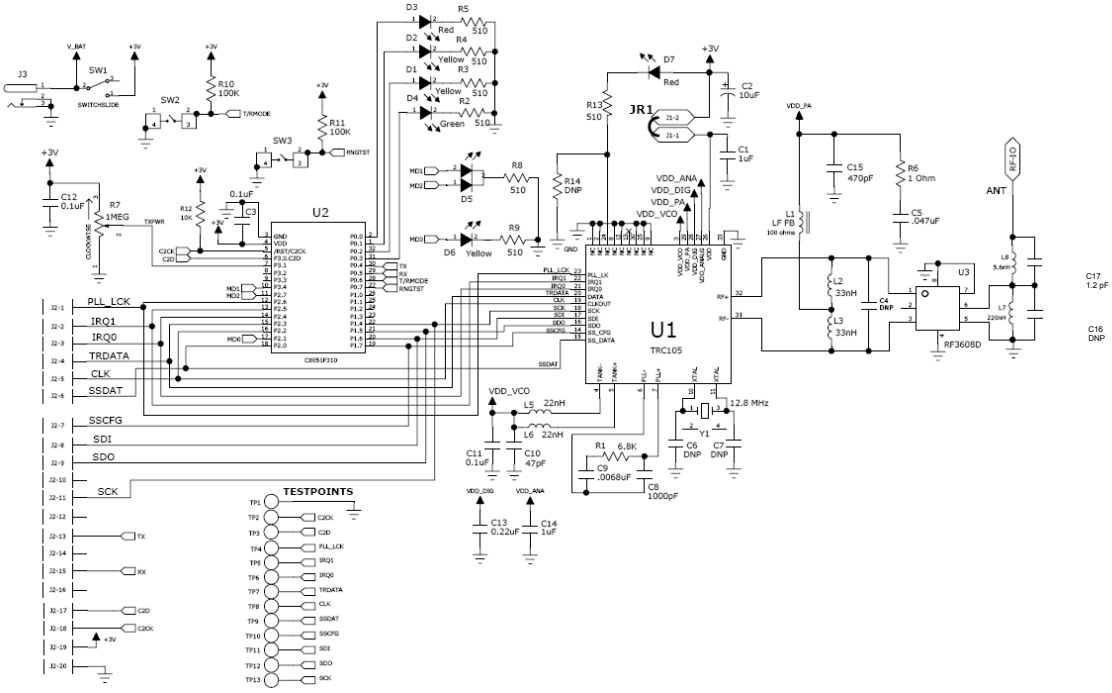



Figure 2

## 2.1 RF Port

The receiver and the transmitter share the same RF pins. Figure 3 shows the implementation of the common front-end. In transmit mode, the PA and the PA regulator are on; the voltage on VDD_PA pin is the nominal voltage of the regulator, about 1.8 V. The external inductances L1 and L4 are used for the PA. In receive mode, both PA and PA regulator are off, and VDD_PA is tied to ground. The external inductances L1 and L4 are used for biasing and matching the LNA, which is implemented as a common gate amplifier.

**Internal RF Port Detail**

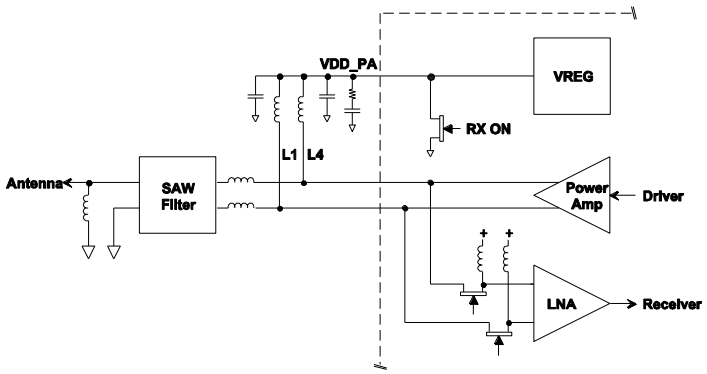



Figure 3

## 2.2 Transmitter

The TRC105 is set to transmit mode when **MCFG00_Chip_Mode[7..5]** bits are set to 100. In Continuous data mode the transmitted data is sent directly to the modulator. The host microcontroller is provided with a bit rate clock by the TRC105 to clock the data; using this clock to send the data synchronously is mandatory in FSK configuration and optional in OOK configuration. In Buffered and Packet data modes the data is first written into the 64-byte FIFO via the SPI interface; data from the FIFO is then used by the modulator.

At the front end of the transmitter, I and Q signals are generated by the base-band circuit which contains a digital waveform generator, two D/A converters and two anti-aliasing low-pass filters. The I and Q signals are two quadrature sinusoids whose frequency is the selected frequency deviation. In FSK mode, the phase shift between I and Q is switched between +90° and -90° according to the input data. The modulation is then performed at this stage, since the information contained in the phase shift will be converted into a frequency shift when the I and Q signals are combined in the first mixers. In OOK mode, the phase shift is kept constant whatever the data. The combination of the I and Q signals in the first mixers creates a fixed frequency signal at a low intermediate frequency which is equal to the selected frequency deviation. After D/A conversion, both I and Q signals are filtered by anti-aliasing filters whose bandwidth is programmed with the register **TXCFG1A_TXInterpfilt[7..4]**. Behind the filters, a set of four mixers combines the I and Q signals and converts them into two I and Q signals at the second intermediate frequency which is equal to 1/8 of the LO frequency, which in turn is equal to 8/9 of the RF frequency. These two new I and Q signals are then combined and up-converted to the desired RF frequency by two quadrature mixers fed by the LO signals. The signal is then amplified by a driver and power amplifier stage.

| MCFG05_PA_Ramp[7..6] | $T_{PA}$ (µs) | Rise/fall (µs) |
|---|---|---|
| 00 | 3 | 2.5/2 |
| 01 | 8.5 | 5/3 |
| 10 | 15 | 10/6 |
| 11 | 23 | 20/10 |

Table 2

OOK modulation is performed by switching on and off the power amplifier and its regulator. The rise and fall times of the OOK signal can be configured in register **MCFG05_PA_Ramp[7..6]**, which controls the charge and discharge time of the regulator. Figure 4 shows the time constants set by **MCFG05_PA_Ramp[7..6].** Table 2 gives typical values of the rise and fall times as defined in Figure 4 when the capacitance connected to the output of the regulator is 0.047 µF.

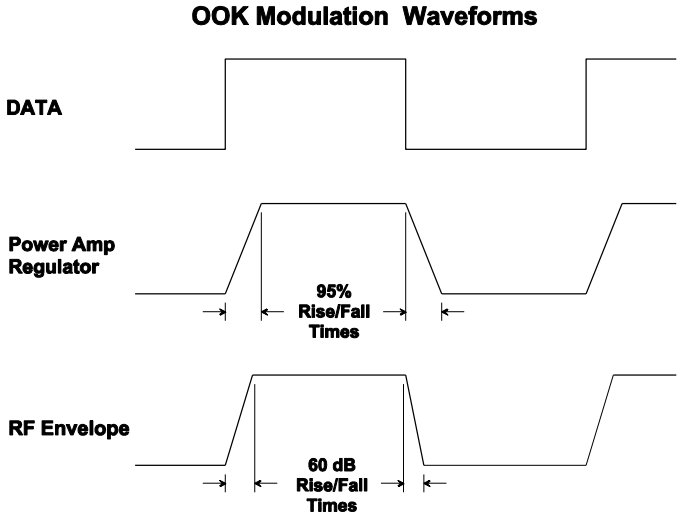


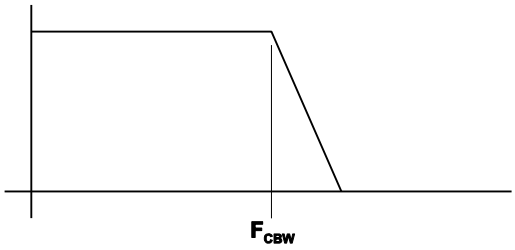

Figure 4

## 2.3 Receiver

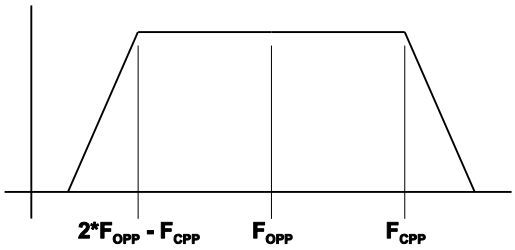
The TRC105 is set to receive mode when **MCFG00_Chip_Mode[7..5]** is set to 011. The receiver is based on a double-conversion architecture. The front-end is composed of an LNA and a mixer whose gains are constant. The mixer down-converts the RF signal to an intermediate frequency which is equal to 1/8 of the LO frequency, which in turn is equal to 8/9 of the RF frequency. Behind this first mixer there is a variable gain IF amplifier that can be programmed from maximum gain to 13.5 dB less in 4.5 dB steps with the **MCFG01_IF_Gain[1..0]** register.

After the variable gain IF amplifier, the signal is down-converted into two I and Q base-band signals by two quadrature mixers which are fed by reference signals at 1/8 the LO frequency. These I and Q signals are then filtered and amplified before demodulation. The first filter is a second-order passive R-C filter whose bandwidth can be programmed to 16 values with the register **RXCFG10_LP_filt[7..4]**. The second filter can be configured as either a third-order Butterworth active filter which acts as a low-pass filter for the zero-IF FSK configuration, or as a polyphase band-pass filter for the low-IF OOK configuration. To select Butterworth low-pass filter operation, bit **RXCFG12_PolyFilt_En[7]** is set to 0. The bandwidth of the Butterworth filter can be programmed to 16 values with the register **RXCFG10_BW_Filt[3..0]**. The low-IF configuration must be used for OOK modulation. This configuration is enabled when the bit **RXCFG12_PolyFilt_En[7]** is set to 1. The center frequency of the polyphase filter can be programmed to 16 values with the register **RXCFG11_PolyFilt[7..4]**. The bandwidth of the filter can be programmed with the register **RXCFG10_BW_Filt[3..0]**. In OOK mode, the value of the low-IF is equal to the deviation frequency defined in register **MCFG02_Freq_dev**. In addition to channel filtering, the function of the polyphase filter is to reject the image. Figure 5 below shows the two configurations of the second IF filter. In the Butterworth configuration, $F_{CBW}$ is the 3 dB cutoff frequency. In the polyphase band-pass configuration $F_{OPP}$ is the center frequency given by **RXCFG11_PolyFilt[7..4]**, and $F_{CPP}$ is the upper 3 dB bandwidth of the filter whose offset, referenced to $F_{OPP}$, is given by **RXCFG10_BW_Filt[3..0].**

## TRC105 Second IF Filter Details




**Butterwoth Low-pass Filter for FSK**




**Polyphase Band-pass Filter for OOK**

Figure 5

After filtering, the I and Q signals are each amplified by a chain of 11 amplifiers having 6 dB of gain each. The outputs of these amplifiers and their intermediate 3 dB nodes are used to evaluate the received signal strength (RSSI). Limiters are located behind the 11 amplifiers of the I and Q chains and the signals at the output of these limiters are used by the FSK demodulator. The RSSI output is used by the OOK demodulator. The global bandwidth of the whole base-band chain is given by the bandwidths of the passive filter, the Butterworth filter, the amplifier chain and the limiter. The maximum achievable global bandwidth when the bandwidths of the first three blocks are programmed at their upper limit is about 350 kHz.

## 2.4 Crystal Oscillator

Crystal specifications for the TRC105 reference oscillator are given in Table 3. Murata recommends the XTL1020P crystal which is specifically designed for use with the TRC105. Note that crystal frequency error will directly trans-late to carrier frequency, bit rate and frequency deviation error.

| Specification | Min | Typical | Max | Units |
|---|---|---|---|---|
| Nominal frequency | - | 12.80000 (fundamental) | - | MHz |
| Load capacitance for Fs | 13.5 | 15 | 16.5 | pF |
| Motional resistance | - | - | 50 | Ω |
| Motional capacitance | 5 | - | 20 | fF |
| Shunt capacitance | 1 | - | 7 | pF |
| Calibration tolerance at 25 °C | - | | ±10 | ppm (1) |
| Stability over temperature range (-40 °C to 85 °C) | 1 | - | ±15 | ppm |
| Aging in first 5 years | - | - | ±2 | ppm/yr |

Table 3

## 2.5 Frequency Synthesizer

The TRC105 VCO operating frequency range is covered in six bands. Operation in each band requires a specific VCO inductor value and configuration parameter setting, as shown in Table 4 below:

| MCFG00_Band[4..2] | Frequency Band MHz |
|---|---|
| 000 | 300-330 |
| 001 | 330-365 |
| 010 | 365-400 |
| 011 | 400-440 |
| 100 | 440-470 |
| 101 | 470-510 |
| 110 to 111 | not used |

Table 4

Each of these bands is divided into four subbands to provide a low phase noise VCO frequency trimming mechanism. Subbands are selected as shown in Table 5 below:

| MCFG00_Subband[1..0] | Subband | 400-440 MHz example |
|---|---|---|
| 00 | 1st quarter | 400-410 |
| 01 | 2nd quarter | 410-420 |
| 10 | 3rd quarter | 420-430 |
| 11 | 4th quarter | 430-440 |

Table 5

Using the VCO as discussed above, the frequency synthesizer generates the local oscillator (LO) signal for the receiver and transmitter sections. The core of the synthesizer is implemented with an integer-N PLL architecture. The frequency is set by three divider parameters R, P and S. R is the frequency divider ratio in the reference frequency path. P and S set the frequency divider ratio in the feedback loop of the PLL. The frequency synthesizer includes a crystal oscillator which provides the frequency reference for the PLL. The equations giving the relationships between the reference crystal frequency, the local oscillator frequency and RF carrier frequency are given below:

$F_{LO} = F_{XTAL}*(75*(P + 1) + S)/(R + 1)$, with P and S in the range 0 to 255, S less than (P + 1), R in the range 64 to 169, and $F_{LO}$ and $F_{XTAL}$ in MHz.

$F_{RF} = 1.125*F_{LO}$, where $F_{RF}$ and $F_{LO}$ are in MHz

$F_{LO}$ is the first local oscillator (VCO) frequency, $F_{XTAL}$ is the reference crystal frequency and $F_{RF}$ is the RF channel frequency. $F_{LO}$ is the frequency used for the first down-conversion of the receiver and the second up-conversion of the transmitter. The intermediate frequency used for the second down-conversion of the receiver and the first up-conversion of the transmitter is equal to 1/8 of $F_{LO}$. As an example, with a crystal frequency of 12.8 MHz and an RF frequency of 434 MHz, $F_{LO}$ is 385.8 MHz and the first IF of the receiver is 48.2 MHz.

There are two sets of divider ratio registers: **SynthR1[7..0]**, **SynthP1[7..0]**, **SynthS1[7..0]**, and **SynthR2[7..0]**, **SynthP2[7..0]**, **SynthS2[7..0]**. The **MCFG05_RF_Freq[0]** bit is used to select which set of registers to use as the current frequency setting. For frequency hopping applications, this reduces the programming and synthesizer settling time when changing frequencies. While the data is being transmitted, the next frequency is programmed and ready. When the current transaction is complete, the **MCFG05_RF_Freq[0]** bit is complemented and the frequency shifts to the next freq according to the contents of the divider ratio registers. This process is repeated for each frequency hop.

## 2.6 PLL Loop Filter

The loop filter for the frequency synthesizer is shown in Figure 6.

**PLL Loop Filter**

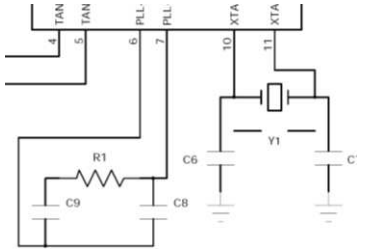



Figure 6

**PLL Loop Filter Components**

| Name | Value | Tolerance |
|---|---|---|
| C8 | 1000 pF | ±10% |
| C9 | 6800 pF | ±10% |
| R1 | 6.8 kΩ | ±5% |

Table 6

Recommended component values for the frequency synthesizer loop filter are provided in Table 6. The loop filter settings are not dependent on the frequency band, so they can be used on all designs. PLL lock status can be provided on Pin 23 by setting the **IRQCFG0E_PLL_LOCK_EN[0]** bit to a 1. Pin 23 goes high when the PLL is locked. The lock status of the PLL can also be checked by reading the **IRQCFG0E_PLL_ LOCK_ST[1]** bit. This bit latches high each time the PLL locks and must be reset by writing a 1 to it.

## 3.0 Operating Modes

The TRC105 has 5 possible chip-level modes. The chip-level mode is set by **MCFG00_Chip_Mode[7..5]**, which is a 3-bit pattern in the configuration register. Table 7 summarizes the chip-level modes:

| MCFG00_Chip_Mode[7..5] | Chip-level Mode | Enabled Functions |
|---|---|---|
| 0 0 0 | Sleep | None |
| 0 0 1 | Standby | Crystal oscillator |
| 0 1 0 | Synthesizer | Crystal and frequency synthesizer |
| 0 1 1 | Receive | Crystal, frequency synthesizer and receiver |
| 1 0 0 | Transmit | Crystal, frequency synthesizer and transmitter |

Table 7

Table 8 gives the state of the digital pins for the different chip-level modes and settings:

| PIN Function | Sleep Mode | Standby Mode | Synthesizer Mode | Receive Mode | Transmit Mode |
|---|---|---|---|---|---|
| nSS_CONFIG* | I | I | I | I | I |
| nSS_DATA* | I | I | I | I | I |
| IRQ0 | TRI | O | O | O | O |
| IRQ1 | TRI | O | O | O | O |
| DATA | TRI | TRI | TRI | O | I |
| CLKOUT | TRI | O | O | O | O |
| SDO** | TRI/O | TRI/O | TRI/O | TRI/O | TRI/O |
| SDI | I | I | I | I | I |
| SCK | I | I | I | I | I |

I = Input, O = Output, TRI = High Impedance
*nSS_CONFIG has priority OVER nSS_DATA
**SDO is an output if nSS_CONFIG = 0 and/or nSS_DATA = 0

Table 8

The TRC105 transmitter and receiver sections support three data handling modes of operation:

- Continuous mode: each bit transmitted or received is accessed directly at the DATA input/output pin.
- Buffered mode: a 64-byte FIFO is used to store each data byte transmitted or received. This data is written to and read from the FIFO through the SPI bus.
- Packet handling mode: in addition to using the FIFO, this data mode builds the complete packet in transmit mode and extracts the useful data from the packet in receive mode. The packet includes a preamble, a start pattern (network address), an optional node address and length byte and the data. Packet data mode can also be configured to perform additional operations like CRC error detection and DC-balanced Manchester encoding or data scrambling.

The buffered and packet handling modes allow the host microcontroller overhead to be significantly reduced. The DATA pin is bidirectional and is used in both transmit and receive modes. In receive mode, DATA represents the demodulated received data. In transmit mode, input data is applied to this pin.

The working length of the FIFO can set to 16, 32, 48 or 64 bytes through the **MCFG0C_FIFO_depth[7..6]** register. In the discussions below describing the FIFO behavior, the explanations are given with an assumption of 64 bytes, but the principle is the same for the four possible FIFO sizes.

The status of the FIFO can be monitored via interrupts which are described in Section 3.7. In addition to the straightforward nFIFOEMPY and FIFOFULL interrupts, additional configurable interrupts Fifo_Int_Tx and Fifo_Int_Rx are also available.

A low-to-high transition occurs on Fifo_Int_Rx when the number of bytes in the FIFO is greater than or equal to the threshold set by **MCFG0C_FIFO_thresh[5..0]** (number of bytes ≥ FIFO_thresh).

A low-to-high transition occurs on Fifo_Int_Tx when the number of bytes in the FIFO is less than or equal to the threshold set by **MCFG0C_FIFO_thresh[5..0]** (number of bytes ≤ FIFO_thresh).

## 3.1 Receiving in Continuous Data Mode

The receiver operates in Continuous data mode when the **MCFG01_Mode[7..6]** bits are set to 00. In this mode, the receiver has two output signals indicating recovered clock, DCLK and recovered NRZ bit DATA. DCLK is connected to output pin IRQ1 and DATA is connected to pin DATA configured in output mode. The data and clock recovery controls the recovered clock signal, DCLK. Data and clock recovery is enabled by **RXCFG12_DCLK_Dis[6]** to 0 (default value). The clock recovered from the incoming data stream appears at DCLK. When data and clock recovery is disabled, the DCLK output is held low and the raw demodulator output appears at DATA. The function of data and clock recovery is to remove glitches from the data stream and to provide a synchronous clock at DCLK. The output DATA is valid at the rising edge of DCLK as shown in Figure 8.

## TRC105 Continuous Mode Demodulation

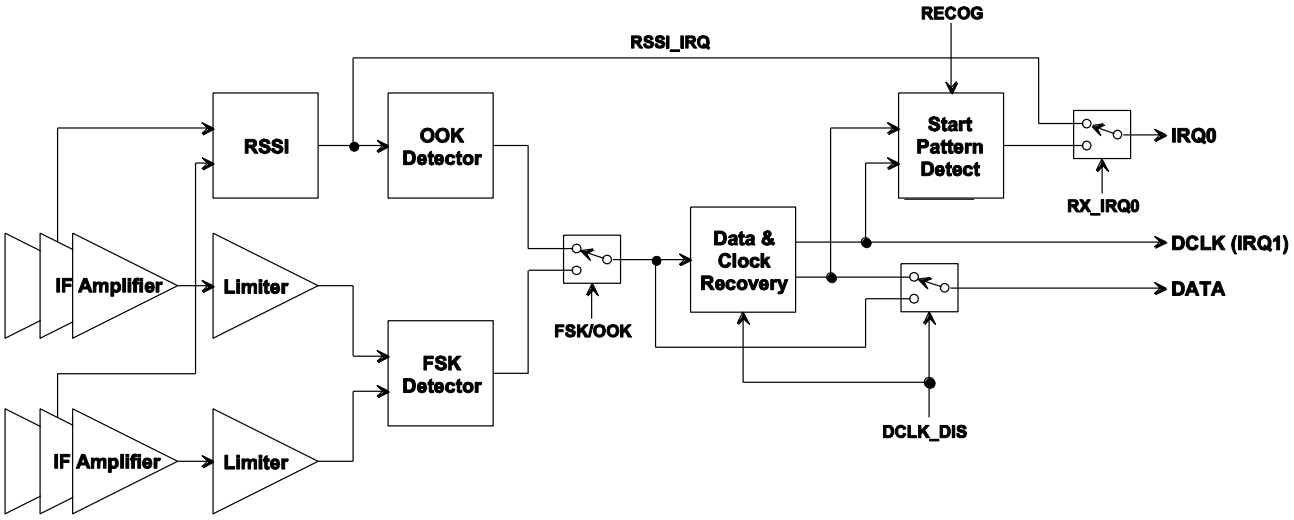



Figure 7

As shown in Figure 7, the demodulator section includes the FSK demodulator, the OOK demodulator, data and clock recovery and the start pattern detection blocks.

If FSK is selected, the demodulation is performed by analyzing the phase between the I and Q limited signals at the output of the base-band channels.

If OOK is selected, the demodulation is performed by comparing the RSSI output value stored in **RXCFG14_ RSSI[7..0]** register to the threshold which can be either a fixed value or a time-variant value depending on the past history of the RSSI output. Table 9 gives the three main possible procedures, which can be selected via the register **MCFG01_RX_OOK[3..2]**:

| Mode | MCFG01_RX_OOK[3..2] | Description |
|---|---|---|
| Fixed Threshold | 00 | RSSI output is compared with a fixed threshold stored in RXCFG13_OOK_thresh |
| Peak | 01 | RSSI output is compared with a threshold which is at a fixed offset below the maximum RSSI. |
| Average | 10 | RSSI output is compared with the average of the last RSSI values. |

Table 9

If the end-user application requires direct access to the output of the demodulator, then the **RXCFG12_DCLK_ Dis[6]** bit is set to 1 disabling the clock recovery. In this case the demodulator output is directly connected to the DATA pin and the IRQ1 pin (DCLK) is set to low.

For proper operation of the TRC105 demodulator in FSK mode, the modulation index β of the input signal should meet the following condition:

$$\beta = \frac{2*F_{DEV}}{BR} \geq 2$$

where $F_{DEV}$ is the frequency deviation in hertz (Hz) and BR is the data rate in bits per second (b/s).

## 3.2 Continuous Mode Data and Clock Recovery

The raw output signal from the demodulator may contain jitter and glitches. Data and clock recovery converts the data output of the demodulator into a glitch-free bit-stream DATA and generates a synchronized clock DCLK to be used for sampling the DATA output as shown in Figure 8. DCLK is available on pin IRQ1 when the TRC105 operates in continuous mode.

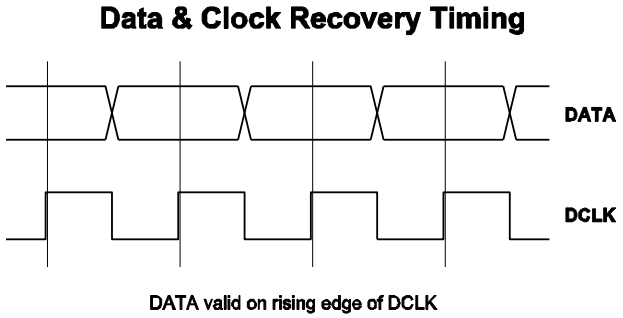



Figure 8

To ensure correct operation of the data and clock recovery circuit, the following conditions have to be satisfied:

- A 1-0-1-0… preamble of at least 24 bits is required for synchronization
- The transmitted bit stream must have at least one transition from 0 to 1 or from 1 to 0 every 8 bits during transmission
- The bit rate accuracy must be better than 2 %.

Data and clock recovery is enabled by default. It is controlled by **RXCFG12_DCLK_Dis[6]**. If data and clock recovery is disabled, the output of the demodulator is directed to DATA and the DCLK output (IRQ1 Pin in continuous mode) is set to 0.

The received bit rate is defined by the values of the **MCFG03 and MCFG04** configuration registers, and is calculated as follows:

$BR = F_{XTAL}/(2*(C + 1)*(D + 1))$, with C in the range of 0 to 255, and D = 31 (suitable for most applications)

with BR the bit rate in kb/s, $F_{XTAL}$ the crystal frequency in kHz, C the value in **MCFG03,** and D the value in **MCFG04.** For example, using a 12.8 MHz crystal (12,800 kHz), the bit rate is 25 kb/s when C = 7 and D = 31.

## 3.3 Continuous Mode Start Pattern Detection

Start pattern detection is activated by setting the **RXCFG12_Recog[5]** bit to 1. The demodulated signal is compared with a pattern stored in the **SYNCFG** registers. The Start Pattern Detect signal (PATTERN) is driven by the output of this comparator and is synchronized by DCLK. It is set to 1 when a pattern match is detected, otherwise set to 0. The pattern detect output is updated at the rising edge of DCLK. The number of bytes used for comparison is defined in the **RXCFG12_Pat_sz[4..3]** register and the number of tolerated bit errors for pattern detection is defined in the **RXCFG12_Ptol[2..1]** register. Figure 9 illustrates the pattern detection process.

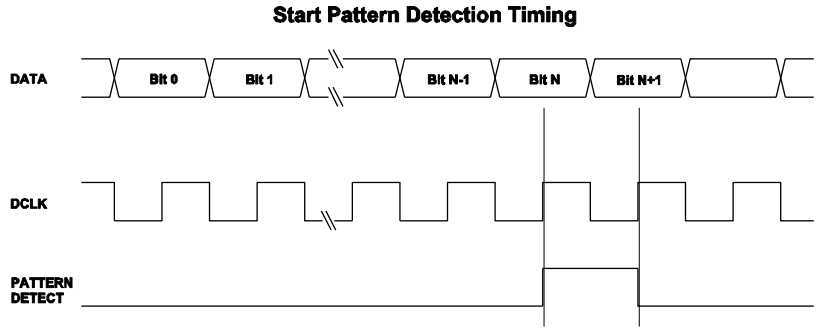



Figure 9

Note that start pattern detection is enabled only if data and clock recovery is enabled.

## 3.4 RSSI

The received signal strength is measured in the amplifier chains behind the second mixers. Each amplifier chain is composed of 11 amplifiers each having a gain of 6 dB and an intermediate output at 3 dB. By monitoring the two outputs of each stage, an estimation of the signal strength with a resolution of 3 dB and a dynamic range of 63 dB is obtained. This estimation is performed 16 times over a period of the I and Q signals and the 16 samples are averaged to obtain a final RSSI value with a 0.5 dB step. The period of the I and Q signal is the inverse of the deviation frequency, which is the low-IF frequency in OOK mode. The RSSI effective dynamic range can be increased to 70 dB by adjusting **MCFG01_IF_Gain[1..0]** for less gain on high signal levels.

The RSSI block can also be used in interrupt mode by setting the bit **IRQCFG0E_RSSI_Int[3]** to 1. When **RXCFG14_RSSI** is equal or greater than a predefined value stored in **IRQCFG0F_RSSI_thld**, bit **IRQCFG0E_ SIG_DETECT[2]** goes high and an interrupt signal RSSI_IRQ is generated on pin IRQ0 if **IRQCFG0D_RX_ IRQ0[7..6]** is set to 01 (see Table 10). The interrupt is cleared by writing a 1 to bit **IRQCFG0E_ SIG_DETECT[2]**. If the bit RSSI_IRQ remains high, the process starts again. Figure 10 shows the timing diagram of RSSI in interrupt mode.

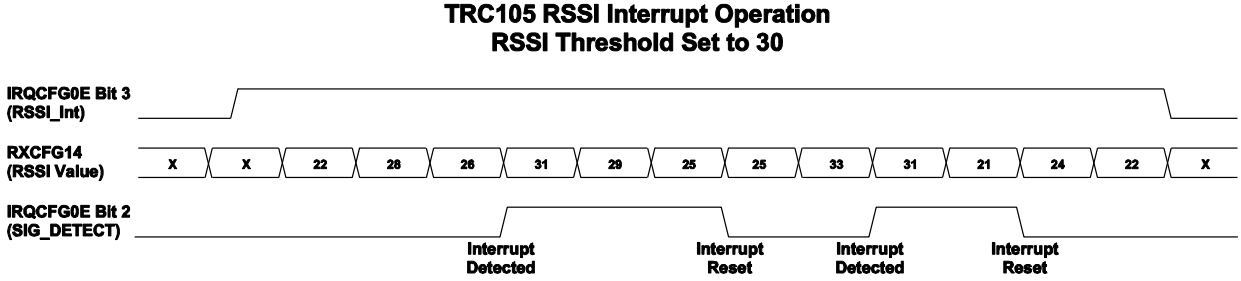



Figure 10

## 3.5 Receiving in Buffered Data Mode

The receiver works in Buffered data mode when the **MCFG01_Mode[7..6]** bits are set to 01. In this mode, the output of the data and clock recovery, i.e., the demodulated and resynchronized signal and the clock signal DCLK are not sent directly to the output pins DATA and IRQ1 (DCLK). These signals are used to store the demodulated data in blocks of 8 bits in a 64-byte FIFO. Figure 11 shows the receiver chain in this mode. The FSK and OOK demodulators, data and clock recovery circuit and start pattern detect block work as described for Continuous data mode, but they are used with two additional blocks, the FIFO and SPI.

**TRC105 Bufferd & Packet Mode Demodulation**

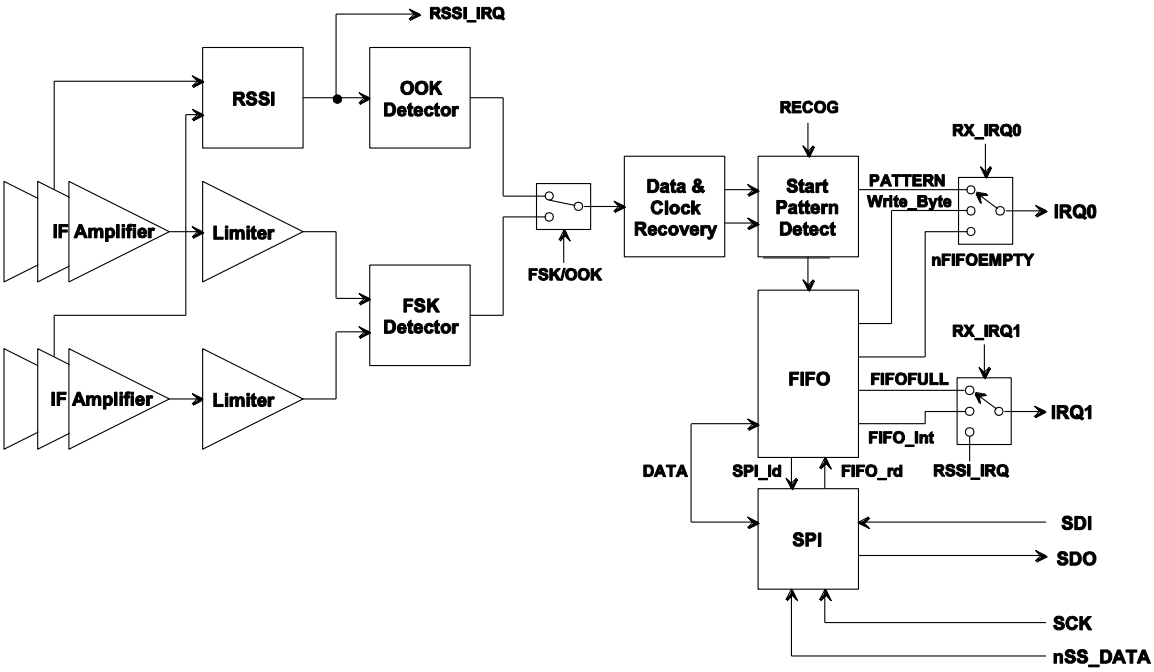



Figure 11

When the TRC105 is in receive mode and **MCFG01_Mode [7..6 ]** bits are set to 01, all of the blocks described above are enabled. In a normal communication frame the data stream is comprised of a 24-bit preamble, a start pattern and data. Upon receipt of a matching start pattern the receiver recognizes the start of data, strips off the preamble and start pattern, and stores the data in the FIFO for retrieval by the host microcontroller. This automated data extraction reduces the loading on the host microcontroller.

The **IRQCFG0E_Start_Fill[7]** bit determines how the FIFO is filled. If **IRQCFG0E_Start_Fill[7]** is set to 0, data only fills the FIFO when a start pattern is detected. Received data bits are shifted into the pattern recognition block which continuously compares the received data with the contents of the **SYNCFG** registers. If a match occurs, the start pattern detect block output is set for one bit period and the **IRQCFG0E_Start_Det[6]** bit is also set. This internal signal can be mapped to the IRQ0 output using interrupt signal mapping. Once a pattern match has occurred, the start pattern detect block will remain inactive until the **IRQCFG0E_Start_Det[6]** bit is reset.

If **IRQCFG0E_Start_Fill[7]** is set to 1, FIFO filling is initiated by asserting **IRQCFG0E_Start_Det[6]**. Once 64 bytes have been written to the FIFO the **IRQCFG0D_FIFOFULL[1]** signal is set. Data should then be read out. If no action is taken, the FIFO will overflow and subsequent data will be lost. If this occurs the **IRQCFG0E_FIFO_ OVR[4]** bit is set to 1. The **IRQCFG0D_FIFOFULL[1]** signal can be mapped to pin IRQ1 as an interrupt for a microcontroller if **IRQCFG0D_RX_IRQ1[5..4]** is set to 01. To recover from an overflow, a 1 must be written to **IRQCFG0D_ FIFO_OVR[4]**. This clears the contents of the FIFO, resets all FIFO status flags and re-initiates pattern detection. Pattern detection can also be re-initiated during a FIFO filling sequence by writing a 1 to **IRQCFG0E_Start_Det[6]**.

The details of the FIFO filling process are shown in Figure 12. As the first byte is written into the FIFO, signal **IRQCFG0D_nFIFOEMPY[0]** is set indicating at least one byte is present. The host microcontroller can then read the contents of the FIFO through the SPI interface. When all data is read from the FIFO, **IRQCFG0D_**

**nFIFOEMPY[0]** is reset. When the last bit of the 64th byte has been written into the FIFO, signal **IRQCFG0D_ FIFOFULL[1]** is set. Data must be read before the next byte is received or it will be overwritten.

The **IRQCFG0D_nFIFOEMPY[0]** signal can be used as an interrupt signal for the host microcontroller by mapping to pin IRQ0 if **IRQCFG0D_RX_IRQ0[7..6]** is set to 10. Alternatively, the WRITE_byte signal may also be used as an interrupt if **IRQCFG0D_RX_IRQ0[7..6]** is set to 01.

Demodulation in buffered mode occurs in the same way as in continuous mode. Received data is directly read from the FIFO and the DATA and DCLK pins are not used. Data and clock recovery in buffered mode is automatically enabled. DCLK is not externally available.

The pattern recognition block is automatically enabled in Buffered data mode. The Start Pattern Detect (PATTERN) signal can be mapped to pin IRQ0. In Buffered data mode RSSI operates the same way as in Continuous data mode. However, RSSI_IRQ may be mapped to IRQ1 instead of to IRQ0 in Continuous data mode.

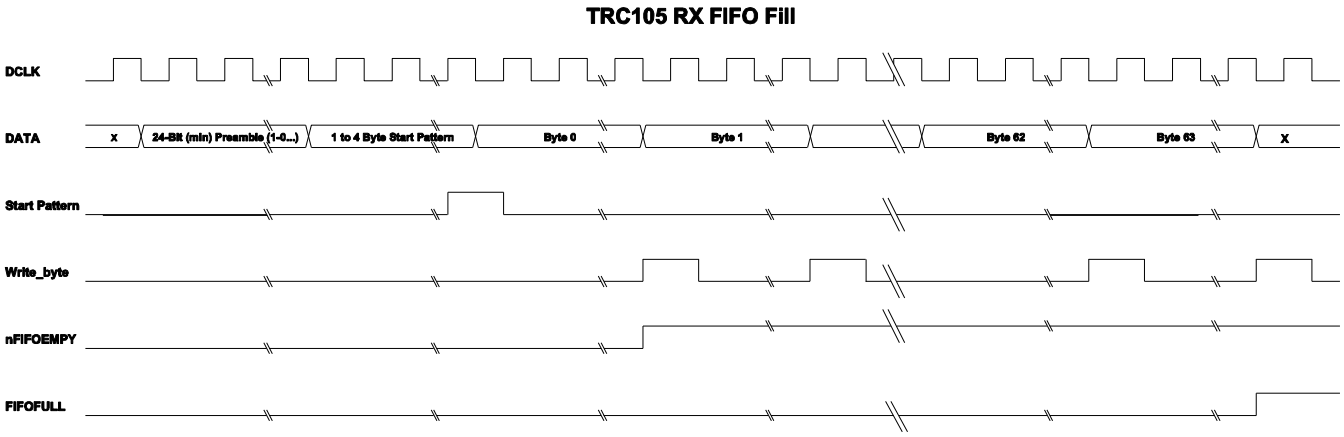



Figure 12

## 3.6 Transmitting in Continuous or Buffered Data Modes

The transmitter operates in Continuous data mode when the **MCFG01_Mode [7..6]** bits are set to 00. Bit clock DCLK is available on pin IRQ1. Bits are clocked into the transmitter on the rising edge of this clock. Data must be stable 2 µs before the rising edge of DCLK and must be held for 2 µs following the rising edge of this clock ($T_{SUDATA}$). To meet this requirement, data can be changed on the falling edge of DCLK. In FSK mode DCLK must be used but is optional in OOK mode.

The transmitter operates in Buffered data mode when the **MCFG01_Mode [5]** bit is set to 1. Data to be transmitted is written to the 64-byte FIFO through the SPI interface. FIFO data is loaded byte-by-byte into a shift register which then transfers the data bit-by-bit to the modulator. FIFO operation in transmit mode is similar to receive mode. Transmission can start immediately after the first byte of data is written into the FIFO or when the FIFO is full, as determined by the **IRQCFG0E_Start_Full[4]** bit setting. If the transmit FIFO is full, the interrupt signal **IRQCFG0D_ FIFOFULL[2]** is asserted on pin IRQ1. If data is written into the FIFO while it is full, the flag **IRQCFG0D_FIFO_OVR[0]** will be set to 1 and the previous FIFO contents will be overwritten. The **IRQCFG0D_ FIFO_OVR[0]** flag is cleared by writing a 1 to it. At the same time the contents of the FIFO are cleared. Once the last data byte in the FIFO is loaded into the shift register driving the transmitter modulator, the flag **IRQCFG0D_ nFIFOEMPY[1]** is set to 0 on pin IRQ0. If new data is not written to the FIFO and the last bit has been transferred to the modulator, the **IRQCFG0E_TX_STOP[5]** bit goes high as the modulator starts to send the last bit. The transmitter must remain on one bit period after **TX_STOP** to transmit the last bit. If the transmitter is switched off (switched to another mode), the transmission stops immediately even if there is still data in the shift register. In

transmit mode the two interrupt signals are IRQ0 and IRQ1. IRQ1 is mapped to **IRQCFG0D_FIFOFULL[2]** signal indicating that the transmission FIFO is full when **IRQCFG0D_TX_IRQ1[3]** is set to 0, or to **IRQCFG0E_TX_ STOP[5]** when **IRQCFG0D_ TX_IRQ1[3]** is set to 1. IRQ0 is mapped to the **IRQCFG0D_nFIFOEMPY[1]** signal. This signal indicates the transmitter FIFO is empty and must be refilled with data to continue transmission.

## 3.7 IRQ0 and IRQ1 Mapping

Two TRC105 outputs are dedicated to host microcontroller interrupts or signaling. The interrupts are IRQ0 and IRQ1 and each have selectable sources. Tables 10, 11, 12 and 13 below summarize the interrupt mapping options. These interrupts are especially useful in Continuous or Buffered data mode operation.

| IRQCFG0D_RX_IRQ0 | Data Mode | IRQ0 | IRQ0 Interrupt Source |
|---|---|---|---|
| 00 | Continuous | Output | Start Pattern Detect |
| 01 | Continuous | Output | RSSI_IRQ |
| 10 | Continuous | Output | Start Pattern Detect |
| 11 | Continuous | Output | Start Pattern Detect |
| 00 | Buffered | Output | None (set to 0) |
| 01 | Buffered | Output | Write_byte |
| 10 | Buffered | Output | nFIFOEMPY |
| 11 | Buffered | Output | Start Pattern Detect |
| 00 | Packet | Output | Data_Rdy |
| 01 | Packet | Output | Write_byte |
| 10 | Packet | Output | nFIFOEMPY |
| 11 | Packet | Output | Node Address Match if ADDRS_cmp is enabled |
| | | | Start Pattern Detect if ADDRS_cmp is disabled |

Table 10

| IRQCFG0D_RX_IRQ1 | Data Mode | IRQ1 | IRQ1 Interrupt Source |
|---|---|---|---|
| 00 | Continuous | Output | DCLK |
| 01 | Continuous | Output | DCLK |
| 10 | Continuous | Output | DCLK |
| 11 | Continuous | Output | DCLK |
| 00 | Buffered | Output | None (set to 0) |
| 01 | Buffered | Output | FIFOFULL |
| 10 | Buffered | Output | RSSI_IRQ |
| 11 | Buffered | Output | FIFO_Int_Rx |
| 00 | Packet | Output | CRC_OK |
| 01 | Packet | Output | FIFOFULL |
| 10 | Packet | Output | RSSI_IRQ |
| 11 | Packet | Output | FIFO_Int_Rx |

Table 11

Tables 12 and 13 describe the interrupts available in transmit mode:

| IRQCFG0D_TX_IRQ0 | Data Mode | IRQ0 | IRQ0 Interrupt Source |
|---|---|---|---|
| 0 | Continuous | Output | None (set to 0) |
| 1 | Continuous | Output | None (set to 0) |
| 0 | Buffered | Output | FIFO_thresh |
| 1 | Buffered | Output | nFIFOEMPY |
| 0 | Packet | Output | FIFO_thresh |
| 1 | Packet | Output | nFIFOEMPY |

Table 12

| IRQCFG0D_TX_IRQ1 | Data Mode | IRQ1 | IRQ0 Interrupt Source |
|---|---|---|---|
| 0 | Continuous | Output | DCLK |
| 1 | Continuous | Output | DCLK |
| 0 | Buffered | Output | FIFOFULL |
| 1 | Buffered | Output | TX_Stop |
| 0 | Packet | Output | FIFOFULL |
| 1 | Packet | Output | TX_Stop |

Table 13

## 3.8 Buffered Clock Output

The buffered clock output is a signal derived from $F_{XTAL}$. It can be used as a reference clock for the host microcontroller and is output on the CLKOUT pin. The **OSCFG1B_Clkout_En[7]** bit controls the CLKOUT pin. When this bit is set to 1, CLKOUT is enabled, otherwise it is disabled. The output frequency of CLKOUT is defined by the value of the **OSCFG1B_Clk_freq[6..2]** parameter which gives the frequency divider ratio applied to $F_{XTAL}$. Refer to Table 42 for programming details. Note: CLKOUT is disabled when the TRC105 is in sleep mode. If sleep mode is used, the host microcontroller must have provisions to run from its own clock source.

## 3.9 Packet Data Mode

The TRC105 provides optional on-chip RX and TX packet handling features. These features ease the development of packet oriented wireless communication protocols and free the MCU resources for other tasks. The options include enabling protocols based on fixed and variable packet lengths, data scrambling, CRC checksum calculations and received packet filtering. All the programmable parameters of the Packet data mode are accessible through the **PKTCFG** configuration registers of the device. The Packet data mode is enabled when the register bit **MCFG01_Mode[7..6 ]** is set to 10 or 11.

The packet handler supports three types of packet formats: fixed length packets, variable length packets, and extended variable length packets. The **PKTCFG1E_Pkt_mode[7]** bit selects either the fixed or the variable length packet formats.

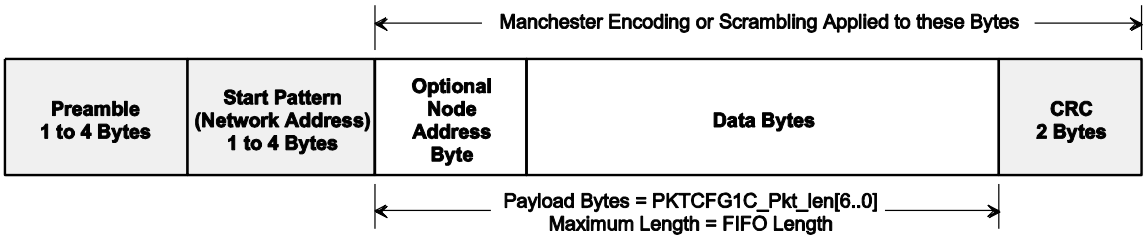
### 3.9.1 Fixed Length Packet Mode

The fixed length packet mode is selected by setting the **PKTCFG1E_Pkt_mode[7]** bit to 0. In this mode the length of the packet is set by the **PKTCFG1C_Pkt_len[6..0]** register up to the size of the FIFO which has been selected.

The length stored in this register is the length of the payload which includes the message data bytes and optional address byte. The fixed length packet format shown in Figure 13 is made up of the following fields:

1. Preamble
2. Start pattern (network address)
3. Node address byte (optional)
4. Data bytes
5. Two-byte CRC checksum (optional)

**Fixed Length Packet Format**




The Preamble, Start Pattern and CRC bytes are added to the packet by the TRC103 during transmit and removed from the packet during receive.

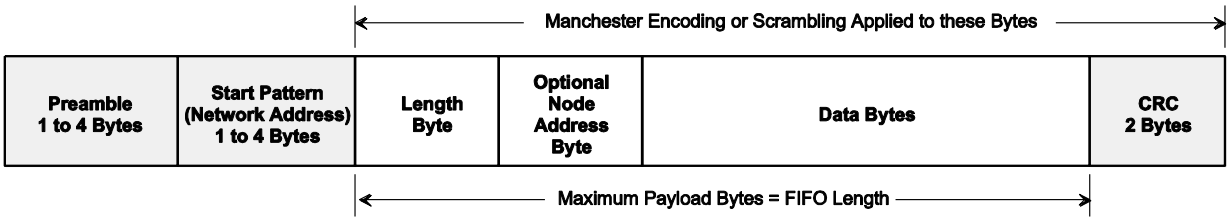
Figure 13

### 3.9.2 Variable Length Packet Mode

The variable length packet mode is selected by setting bit **PKTCFG1E_Pkt_mode[7]** to 1. The packet format shown in Figure 14 is programmable and is made up of the following fields:

1. Preamble
2. Start pattern (network address)
3. Length byte
4. Node address byte (optional)
5. Data bytes
6. Two-byte CRC checksum (optional)

**Variable Length Packet Format**




The Preamble, Start Pattern and CRC bytes are added to the packet by the TRC103 during transmit and removed from the packet during receive.

Figure 14

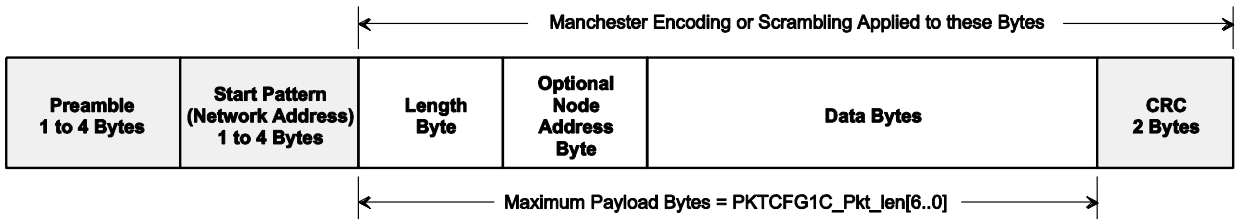
In variable length packet mode, the length of the rest of the payload is given by the first byte written to the FIFO. The length byte itself is not included in this count. The **PKTCFG1C_Pkt_len[6..0]** parameter is used to set the maximum received payload length allowed. Any received packet having a value in the length byte greater than this maximum is discarded. The variable length packet format accommodates payloads, including the length byte, up to the length of the FIFO.

### 3.9.3 Extended Variable Length Packet Mode

The extended variable length packet mode is selected by setting bit **PKTCFG1E_Pkt_mode[7]** to 1 and setting **PKTCFG1C_Pkt_len[6..0]** to a value between 65 and 127. The packet format shown in Figure 15 is programmable and is made up of the following fields:

1. Preamble
2. Start pattern (network address)
3. Length byte
4. Node address byte (optional)
5. Data bytes
6. Two-byte CRC checksum (optional)

**Extended Variable Length Packet Format, 65 to 127 Bytes**




The Preamble, Start Pattern and CRC bytes are added to the packet by the TRC103 during transmit and removed from the packet during receive.

Figure 15

In extended variable length packet mode, the length of the rest of the payload is given by the first byte written to the FIFO. The length byte itself is not included in this count. There are a number of ways to use the extended variable length packet capability. The most common way is outlined below:

1. Set **PKTCFG1C_Pkt_len[6..0]** to a value between 65 (0x41) and 127 (0x7F). This sets the maximum allowed payload in extended packet mode. Any received packet having a value in the length byte greater than this maximum is discarded.

2. Set **PKTCFG1E_Pkt_mode[7**] to 1 for variable length packet mode operation. Set the **PKTCFG1E_ Preamb_len[6..5]** bits to 10 or 11 for a 3 or 4 byte preamble. Set the **PKTCFG1E_CRC_En[3]** bit to 1 to enable CRC processing. Set the **PKTCFG1E_Pkt_ADDRS_cmp[2..1]** bits as required. Clear the **PKTCFG1E_ CRC_stat[0]** bit by writing a 1 to it.

3. Set **MCFG0C_FIFO_depth[7..6]** bits to 11 for a 64 byte FIFO length.

4. Set the **MCFG0C_FIFO_thresh[5..0]** to approximately 31(0x1F). This sets the threshold to 32, near the mid point of the FIFO. Provided the host microcontroller is relatively fast (usual case), this setting can be used for monitoring the FIFO in both transmit and receive. If the host microcontroller is relatively slow, set the threshold to a value lower than 31 for receive, and higher than 31 for transmit.

5. Set the **IRQCFG0D_RX_IRQ1[5..4]** bits to 11. This maps FIFO_Int_Rx interrupt to IRQ1, which trips when the number of received bytes in the FIFO is equal to or greater than the value in MCFG05_FIFO_thresh. IRQ1 will then signal received bytes must be retrieved. If received bytes are not retrieved before the FIFO completely fills, data will be lost.

6. Set the **IRQCFG0D_TX_IRQ0[3]** bit to 0. This causes a transmission to start when the number of transmit bytes in the FIFO is equal to or greater than the value in MCFG0C_FIFO_thresh. Also, the FIFO_Int_Tx interrupt is mapped to IRQ0 in transmit mode, and is set when the number of bytes in the FIFO is equal to or less than the value in MCFG0C_FIFO_thresh. IRQ0 will then signal more bytes can be added to the FIFO. If more message bytes are not added in time, the transmission will cease prematurely and data will be lost. Likewise, if more bytes are sent to the FIFO than it has room for, data will be lost.

7. When receiving an extended variable length packet, monitor IRQ1. When IRQ1 trips, clock out some of the received bytes from the FIFO (leave at least one byte in the FIFO). Repeat the partial packet retrieval each time IRQ1 triggers. The first byte received is the number of message bytes, and can be used to tell when the last message byte has been retrieved. When it is determined that the remaining message bytes will not overflow the FIFO, the **IRQCFG0D_RX_IRQ1[5..4]** bits can be set to 00, which maps CRC_OK to IRQ1. After the CRC is checked, the final bytes can be read from the FIFO and the **IRQCFG0D_RX_IRQ1[5..4**] bits can be reset to 11 to track

FIFO_Int_Rx when the next packet is received. Note that CRC mapping to IRQ1 is not required if the CRC state is read from the **PKTCFG1E_ CRC_stat[0]** bit prior to reading the final FIFO bytes.

8. When transmitting an extended variable length packet, begin filling the FIFO until IRQ0 trips, indicating the FIFO is half full. Add up to 32 bytes to the FIFO (64 - (MCFG05_ FIFO_thresh +1)) when IRQ0 resets. Repeat the partial packet loading each time IRQ0 resets until all bytes to be transmitted have been clocked in. The **IRQCFG0D_TX_IRQ1[3**] bit can then be set to 1, which allows the TX_STOP event to be mapped to IRQ1. TX_STOP signals the last bit to be transmitted has been transferred the modulator. Allow one bit period for this bit to be transmitted before switching out of transmit mode.

### 3.9.4 Packet Payload Processing in Transmit and Receive

The TRC105 packet handler constructs transmit packets using the payload in the FIFO. In receive, it processes the packets and extracts the payload to the FIFO. Packet processing in transmit and receive are detailed below.

For transmit, the packet handler adds the following fields and processing to the payload in the FIFO:

1. One automatic preamble byte
2. One to four additional preamble bytes, programmable and usually set to 3 or 4 bytes
3. One to four start pattern bytes, programmable and usually set to at least 2 bytes
4. Optional CRC checksum calculated over the FIFO payload and appending to the end of the packet
5. Optional Manchester encoding or DC-balanced scrambling

The payload in the FIFO may contain one or both of the following optional fields:

1. A length byte if the variable packet length mode is selected
2. A node address byte

If the FIFO is filled while transmit mode is enabled, and if **IRQCFG0E_Start_Full[4]** is set to 1, the modulator waits until the first byte is written into the FIFO, then it starts sending the programmed preamble bytes followed by the start pattern and the user payload. If **IRQCFG0E_Start_Full[4 ]** is set to 0 in the same conditions, the modulator waits until the number of bytes written in the FIFO is equal to the number defined in the register **MCFG05_ FIFO_thresh[5..0]**. Note that the transmitter automatically sends preamble bytes in addition the number programmed while in transmit mode and waiting for the FIFO to receive the required number of bytes to start data transmission. Data to be transmitted can also be written into the FIFO during standby mode. In this case, the data is automatically transmitted when the transmit mode is enabled and the transmitter reaches its steady state.

If CRC is enabled, the CRC checksum is calculated over the payload bytes. This 16-bit checksum is sent after the bytes in the FIFO. If CRC is enabled, the TX_STOP bit is set when the last CRC bit is transferred to the TX modulator. If CRC is not enabled, the TX_STOP bit is set when the last bit from the FIFO is transferred to the TX modulator. Note that the transmitter must remain on one bit period after the TX_STOP bit is set while the last bit is being transmitted. If the transmitter remains on following the transmission of the last bit after TX_STOP is set, the transmitter will send preamble bytes. If Manchester encoding or scrambling is enabled, all data except the preamble and start pattern is encoded or scrambled before transmission. Note that the length byte in the FIFO determines the length of the packet to be sent and the **PKTCFG1C_Pkt_len[6..0]** parameter is not used in transmit.

In receive, the packet handler retrieves the payload by performing the following steps:

1. Data and clock recovery synchronization to the preamble
2. Start pattern detection
3. Optional address byte check
4. Error detection through CRC

When receive mode is enabled, the demodulator detects the preamble followed by the start pattern. If fixed length packet format is enabled, then the number of bytes received as the payload is given by the **PKTCFG1C_Pkt_ len[6..0]** parameter. In variable length and extended variable length packet modes, the first byte received after the start pattern is interpreted as the length of the balance of the payload. An internal length counter is initialized to this length. The **PKTCFG1C_Pkt_len[6..0]** register must be set to a value which is equal to or greater than the maximum expected length byte value of the received packet. If the length byte value of a received packet is greater than the value in the **PKTCFG1C_ Pkt_len[6..0]** register, the packet is discarded. Otherwise the packet payload begins loading into the FIFO.

If address match is enabled, the second byte received in a variable length mode or the first byte in the fixed length mode is interpreted as the node address. If this address matches the byte in **PKTCFG1D_Node_Addrs[7..0]**, reception of the packet continues, otherwise it is stopped. A CRC check is performed if **PKTCFG1E_CRC_ En[3]** is set to 1. If the CRC check is successful, a 1 is loaded in the **PKTCFG1E_CRC_stat[0]** bit, and CRC_OK and Dat_Rdy interrupts are simultaneously generated on IRQ1 and IRQ0 respectively. This signals that the payload or balance of the payload can be read from the FIFO. In receive mode, address match, Dat_Rdy, and CRC_OK interrupts and the CRC_stat bit are reset when the last byte in the FIFO is read. Note the FIFO can be read in standby mode by setting **PGCFG1F_ RnW_FIFO[6]** bit to 1. In standby, reading the last FIFO byte does not clear CRC_OK and the CRC_stat bit. They are reset once the TRC103 is put in receive mode again and a start pattern is detected.

If the CRC check fails, the FIFO is cleared and no interrupts are generated. This action can be overridden by setting **PGCFG1F_CRCclr_auto[7**] to 1, which forces a Data_Rdy interrupt and preserves the payload in the FIFO even if the CRC fails.

If address checking is enabled, the second byte received in a variable length mode or the first byte in the fixed length mode is interpreted as the node address. If this address matches the byte in **PKTCFG1D_Node_ Addrs[7..0]**, reception of the packet continues, otherwise it is stopped. A CRC check is performed if **PKTCFG1E_ CRC_En[3]** is set to 1. If the CRC check is successful, a 1 is loaded in the **PKTCFG1E_CRC_stat[0]** bit, and CRC_OK and Dat_Rdy interrupts are simultaneously generated on IRQ1 and IRQ0 respectively, signaling the payload or balance of the payload can be read from the FIFO. Note the FIFO can be read in standby mode by setting **PGCFG1F_ RnW_FIFO[6]** bit to 1. If the CRC check fails, the FIFO is cleared and no interrupts are generated. This action can be overridden by setting **PGCFG1F_CRCclr_auto[7**] to 1, which forces a Data_Rdy interrupt and preserves the payload in the FIFO even if the CRC fails.

### 3.9.5 Packet Filtering

Received packets can be filtered based on two criteria: length filtering and address filtering. In variable length or extended variable length packet formats, **PKTCFG1C_Pkt_len[6..0]** stores the maximum packet length permitted. If the received packet length is greater than this value, then the packet is discarded. Node address filtering is enabled by setting parameter **PKTCFG1E_Addrs_cmp[2..1]** to any value other than 00, i.e., 01, 10 or 11. These settings enable the following three options:

**PKTCFG1E_Addrs_cmp[2..1]** = 01: This configuration activates the address filtering function on the packet handler and the received address byte is compared with the address in the **PKTCFG1D_Node_Addrs[7..0]** register. If both address bytes are the same, the received packet is for the current destination and is stored in FIFO. Otherwise it is discarded. An interrupt can also be generated on IRQ0 if the address comparison is successful.

**PKTCFG1E_Addrs_cmp[2..1]** = 10: In this configuration the received address is compared to both the **PKTCFG1D_Node_Addrs[7..0]** register and constant 0x00. If the received address byte matches either value, the packet is accepted. An interrupt can also be generated on IRQ0 if the address comparison is successful. The 0x00 address is useful for sending broadcast packets.

**PKTCFG1E_Addrs_cmp[2..1]** = 11: In this configuration the packet is accepted if the received address matches either the **PKTCFG1D_Node_addrs[7..0]** register, 0x00 or 0xFF. An interrupt can also be generated on IRQ0 if the address comparison is successful. The 0x00 and 0xFF addresses are useful for sending two types of broadcast packets.

### 3.9.6 Cyclic Redundancy Check

CRC check is enabled by setting the **PKTCFG1E_CRC_En[3]** bit to 1. A 16-bit CRC checksum is calculated on the payload part of the packet and is appended to the end of the transmitted message. The CRC checksum is calculated on the received payload and compared to the transmitted CRC. The result of the comparison is stored in the **PKTCFG1E_CRC_stat[0]** bit and an interrupt can also be generated on IRQ1. The CRC is based on the CCITT polynomial as shown in Figure 16. The CRC also detects errors due to leading and trailing zeros.

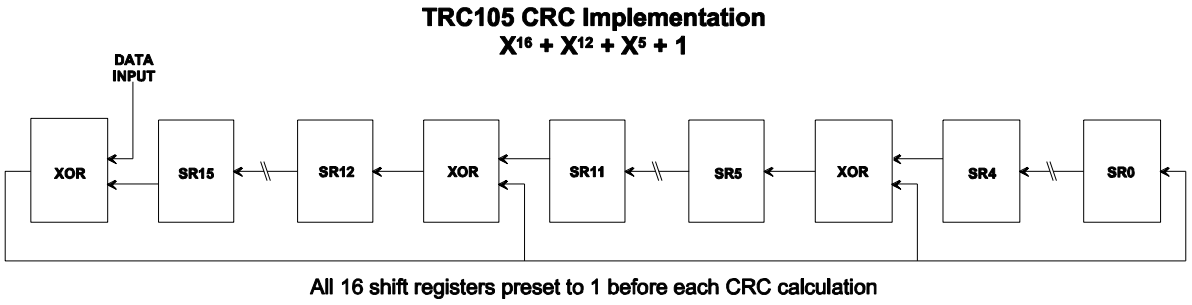



Figure 16

### 3.9.7 Manchester Encoding

Manchester encoding is enabled by setting the **PKTCFG1C_Man_en[7]** bit to 1, and can only be used in Packet data mode. Figure 17 illustrates Manchester encoding. NRZ data is converted to Manchester by encoding 1 bits as 10 chip sequences, and 0 bits as 01 chip sequences. Manchester encoding guarantees DC-balance and frequent data transitions in the encoded data. Note the maximum Manchester chip rate corresponds to the maximum bit rate given in the specifications in Table 52.

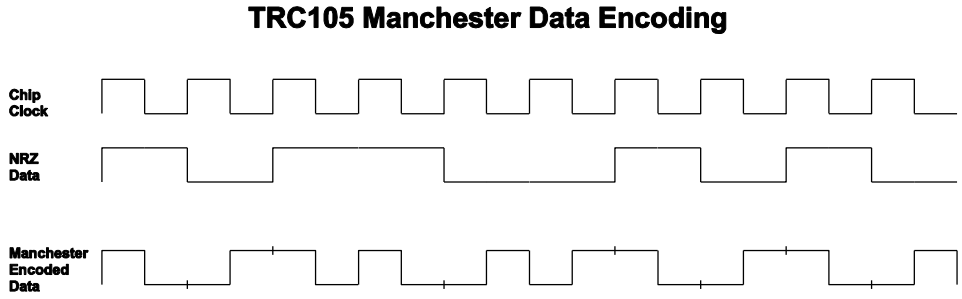



Figure17

In transmit, Manchester encoding is applied only to the payload and CRC parts of the packet. The receiver decodes the payload and CRC before performing other packet processing tasks.

### 3.9.8 DC-Balanced Scrambling

A payload may contain long sequences of 1 or 0 bits. These sequences would introduce DC biases in the transmitted signal, causing a non-uniform power distribution spectrum. These sequences would also degrade the performance of the demodulation and data and clock recovery functions in the receiver. System performance can be enhanced if the payload bits are randomized to reduce DC biases and increase the number of bit transitions.

As discussed above, DC-balanced data can be obtained by using Manchester encoding, which ensures that there are no more than two consecutive 1's or 0's in the transmitted data. However, this reduces the effective bit-rate of the system because it doubles the amount of data to be transmitted.

Another technique called scrambling (whitening) is widely used for randomizing data before radio transmission. The data is scrambled using a random sequence on the transmit side and then descrambled on the receive side using the same sequence.

The TRC105 packet handler provides a mechanism for scrambling the packet payload. A 9-bit LFSR is used to generate a random sequence. The payload and the 16-bit CRC checksum are XOR'd with this random sequence as shown in Figure 18. The data is descrambled on the receiver side by XORing with the same random sequence. The scrambling/descrambling process is enabled by setting the **PKTCFG1E_Scrmb_En[4]** bit to 1.

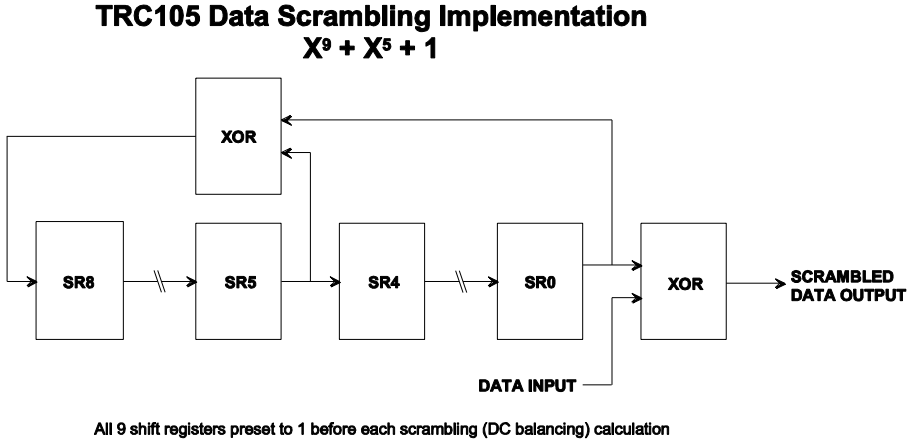



Figure 18

## 3.10 SPI Configuration Interface

The TRC105 contains two SPI-compatible interfaces, one to read and write the configuration registers, the other to read and write FIFO data. Both interfaces are configured in slave mode and share the same pins: SDO (SPI Slave Data Out), SDI (SPI Slave Data In), and SCK (Serial Clock). Two pins are provided to select the SPI connection. The nSS_CONFIG pin allows access to the configuration registers and the nSS_DATA pin allows access to the FIFO. Figure 19 shows a typical connection between a host microcontroller and the SPI interface.