



# nRF31512

2.4 GHz RF SoC

## Product Specification v1.0

### Key Features

- nRF24L01+ compatible 2.4 GHz transceiver (250 kbps, 1 Mbps, and 2 Mbps air data rates)
- Fast 8051 compatible microcontroller
- 17 kB program memory (on-chip OTP)
- 512 bytes data memory (on-chip XRAM)
- 128 bit AES encryption HW accelerator
- 8 to 10 bit ADC
- High flexibility I/Os
- Supply range: 1.9 V - 3.3 V
- Temperature range: -10 Celsius - +60 Celsius
- RoHS compliant versions in packages:
- nRF31512 in 5x5 mm QFN32

### Applications

- Computer peripherals:
  - Mouse
  - Remote controls
  - Simple sensor systems
  - Active RFID
  - Simple Remote controlled toys

## Liability disclaimer

Nordic Semiconductor ASA reserves the right to make changes without further notice to the product to improve reliability, function or design. Nordic Semiconductor ASA does not assume any liability arising out of the application or use of any product or circuits described herein.

All application information is advisory and does not form part of the specification.

## Limiting values

Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the specifications are not implied. Exposure to limiting values for extended periods may affect device reliability.

## Life support applications

Nordic Semiconductor's products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Nordic Semiconductor ASA customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nordic Semiconductor ASA for any damages resulting from such improper use or sale.

### Data sheet status

Objective product specification	This product specification contains target specifications for product development.
Preliminary product specification	This product specification contains preliminary data; supplementary data may be published from Nordic Semiconductor ASA later.
Product specification	This product specification contains final product specifications. Nordic Semiconductor ASA reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.

## Contact details

For your nearest dealer, please see <http://www.nordicsemi.com>.

Information regarding product updates, downloads, and technical support can be accessed through your My Page account on our homepage.

**Main office:** Otto Nielsens veg 12  
 7052 Trondheim  
 Norway  
 Phone: +47 72 89 89 00  
 Fax: +47 72 89 89 89

**Mailing address:** Nordic Semiconductor  
 P.O. Box 2336  
 7004 Trondheim  
 Norway



## Revision History

Date	Version	Description
January 2013	1.0	First release of Product Specification.

## RoHS and REACH statement

Nordic Semiconductor's products meet the requirements of Directive 2002/95/EC of the European Parliament and of the Council on the Restriction of Hazardous Substances (RoHS) and the requirements of the REACH regulation (EC 1907/2006) on Registration, Evaluation, Authorization and Restriction of Chemicals. The SVHC (Substances of Very High Concern) candidate list is continually being updated. Complete hazardous substance reports, material composition reports and latest version of Nordic's REACH statement can be found on our website <http://www.nordicsemi.com>.

## Contents

<b>1</b>	<b>Introduction.....</b>	<b>8</b>
1.1	Prerequisites.....	8
1.2	Writing conventions.....	8
<b>2</b>	<b>Product overview.....</b>	<b>9</b>
2.1	Features.....	9
2.2	Block diagram.....	10
2.3	Pin assignments.....	11
2.3.1	nRF31512 QFN32 package.....	11
2.4	Pin functions.....	12
<b>3</b>	<b>RF transceiver.....</b>	<b>13</b>
3.1	Features.....	13
3.2	Block diagram.....	14
3.3	Functional description.....	14
3.3.1	Operational Modes.....	14
3.3.2	Air data rate.....	19
3.3.3	RF channel frequency.....	19
3.3.4	Received Power Detector measurements.....	20
3.3.5	PA control.....	20
3.3.6	RX/TX control.....	20
3.4	Enhanced ShockBurst™.....	21
3.4.1	Features.....	21
3.4.2	Enhanced ShockBurst™ overview.....	21
3.4.3	Enhanced Shockburst™ packet format.....	22
3.4.4	Automatic packet assembly.....	24
3.4.5	Automatic packet disassembly.....	25
3.4.6	Automatic packet transaction handling.....	26
3.4.7	Enhanced ShockBurst™ flowcharts.....	28
3.4.8	MultiCeiver™.....	31
3.4.9	Enhanced ShockBurst™ timing.....	33
3.4.10	Enhanced ShockBurst™ transaction diagram.....	36
3.4.11	Compatibility with ShockBurst™.....	40
3.5	Data and control interface.....	41
3.5.1	SFR registers.....	41
3.5.2	SPI operation.....	42
3.5.3	Data FIFO.....	44
3.5.4	Interrupt.....	44
3.6	Register map.....	46
3.6.1	Register map table.....	46
<b>4</b>	<b>MCU.....</b>	<b>52</b>
4.1	Block diagram.....	53
4.2	Features.....	53
4.3	Functional description.....	54
4.3.1	Arithmetic Logic Unit (ALU).....	54
4.3.2	Instruction set summary.....	55
4.3.3	Opcode map.....	60
<b>5</b>	<b>Memory and I/O organization.....</b>	<b>63</b>

5.1	System space.....	64
5.2	PDATA memory addressing.....	64
5.3	MCU Special Function Registers .....	65
5.3.1	Accumulator - ACC .....	65
5.3.2	B Register – B .....	65
5.3.3	Program Status Word Register - PSW .....	65
5.3.4	Stack Pointer – SP .....	66
5.3.5	Data Pointer – DPH, DPL.....	66
5.3.6	Data Pointer 1 – DPH1, DPL1 .....	66
5.3.7	Data Pointer Select Register – DPS.....	67
5.3.8	PCON register .....	67
5.3.9	Special Function Register Map.....	67
5.3.10	Special Function Registers reset values .....	69
<b>6</b>	<b>OTP memory .....</b>	<b>71</b>
6.1	Features.....	71
6.2	Block diagram .....	71
6.3	Functional description.....	72
6.3.1	OTP memory configuration SFR .....	72
6.3.2	Memory compatibility with flash version of nRF24LE1 .....	72
6.3.3	Brown-out.....	72
6.3.4	Temperature restriction for OTP programming .....	73
6.3.5	OTP programming from the MCU .....	73
6.3.6	OTP programming through SPI.....	73
<b>7</b>	<b>Random Access memory (RAM) .....</b>	<b>78</b>
7.1	SRAM configuration.....	78
<b>8</b>	<b>Timers/counters .....</b>	<b>80</b>
8.1	Features.....	80
8.2	Block diagram .....	80
8.3	Functional description.....	81
8.3.1	Timer 0 and Timer 1 .....	81
8.3.2	Timer 2 .....	83
8.4	SFR registers .....	85
8.4.1	Timer/Counter control register – TCON.....	85
8.4.2	Timer mode register - TMOD .....	86
8.4.3	Timer 0 – TH0, TL0 .....	86
8.4.4	Timer 1 – TH1, TL1 .....	86
8.4.5	Timer 2 control register – T2CON.....	87
8.4.6	Timer 2 – TH2, TL2 .....	87
8.4.7	Compare/Capture enable register – CCEN .....	88
8.4.8	Capture registers – CC1, CC2, CC3 .....	88
8.4.9	Compare/Reload/Capture register – CRCH, CRCL.....	89
8.5	Real Time Clock - RTC .....	89
8.5.1	Features.....	89
8.5.2	Functional description of SFR registers .....	89
<b>9</b>	<b>Interrupts .....</b>	<b>92</b>
9.1	Features.....	92
9.2	Block diagram .....	92
9.3	Functional description.....	93

9.4	SFR registers .....	93
9.4.1	Interrupt Enable 0 Register – IEN0 .....	94
9.4.2	Interrupt Enable 1 Register – IEN1 .....	94
9.4.3	Interrupt Priority Registers – IP0, IP1 .....	95
9.4.4	Interrupt Request Control Registers – IRCON .....	96
<b>10</b>	<b>Watchdog .....</b>	<b>97</b>
10.1	Features .....	97
10.2	Block diagram .....	97
10.3	Functional description .....	97
<b>11</b>	<b>Power and clock management .....</b>	<b>99</b>
11.1	Block diagram .....	99
11.2	Modes of operation .....	99
11.3	Functional description .....	103
11.3.1	Clock control .....	103
11.3.2	Power down control – PWRDWN .....	106
11.3.3	Operational mode control - OPMCON .....	106
11.3.4	Reset result – RSTREAS .....	107
11.3.5	Wakeup configuration register – WUCON .....	107
11.3.6	Pin wakeup configuration .....	108
<b>12</b>	<b>Power supply supervisor .....</b>	<b>109</b>
12.1	Features .....	109
12.2	Block diagram .....	109
12.3	Functional description .....	109
12.3.1	Power-on reset .....	109
12.3.2	Brown-out reset .....	110
12.3.3	Power-fail comparator .....	110
12.4	SFR registers .....	111
<b>13</b>	<b>On-chip oscillators .....</b>	<b>112</b>
13.1	Features .....	112
13.2	Block diagrams .....	112
13.3	Functional description .....	112
13.3.1	16 MHz crystal oscillator .....	112
13.3.2	16 MHz RC oscillator .....	113
13.3.3	External 16 MHz clock .....	113
13.3.4	32.768 kHz RC oscillator .....	114
13.3.5	Synthesized 32.768 kHz clock .....	114
13.3.6	External 32.768 kHz clock .....	114
<b>14</b>	<b>Encryption/decryption accelerator .....</b>	<b>115</b>
14.1	Features .....	115
14.2	Block diagram .....	115
14.3	Functional description .....	115
<b>15</b>	<b>General purpose I/O port and pin assignments .....</b>	<b>117</b>
15.1	Block diagram .....	117
15.2	Functional description .....	118
15.2.1	General purpose I/O pin functionality .....	118
15.2.2	PortCrossbar functionality .....	119
15.3	I/O pin maps .....	120

15.3.1	Pin assignments for nRF31512 in 32 pin package .....	121
15.3.2	Programmable registers.....	121
<b>16</b>	<b>SPI .....</b>	<b>126</b>
16.1	Features.....	126
16.2	Block diagram .....	126
16.3	Functional description.....	127
16.3.1	SPI master .....	127
16.3.2	SPI timing.....	128
<b>17</b>	<b>ADC.....</b>	<b>132</b>
17.1	Features.....	132
17.2	Functional description.....	132
17.2.1	Activation.....	132
17.2.2	Input selection .....	132
17.2.3	Reference .....	133
17.2.4	Resolution .....	133
17.2.5	Conversion modes.....	133
17.2.6	Output data coding .....	133
17.2.7	SFR registers.....	134
<b>18</b>	<b>Absolute maximum ratings .....</b>	<b>136</b>
<b>19</b>	<b>Operating condition.....</b>	<b>137</b>
<b>20</b>	<b>Electrical specifications .....</b>	<b>138</b>
20.1	OTP memory .....	141
20.2	Power consumption .....	142
<b>21</b>	<b>Mechanical specifications .....</b>	<b>143</b>
<b>22</b>	<b>Reference circuits .....</b>	<b>144</b>
22.1	nRF31512, 5x5 mm QFN32.....	144
22.1.1	Schematic .....	144
22.1.2	Layout .....	145
22.1.3	Bill Of Materials .....	145
<b>23</b>	<b>Ordering information .....</b>	<b>146</b>
23.1	Package marking .....	146
23.1.1	Abbreviations.....	146
23.2	Ordering codes.....	147
<b>24</b>	<b>Glossary .....</b>	<b>148</b>

# 1 Introduction

nRF31512 is a member of the nRF31 series of entry level 2.4 GHz SoC devices. They combine a 2.4 GHz radio, a fast 8051 compatible CPU, essential peripherals, and on-air compatibility with the other product series from Nordic Semiconductor. The nRF31 series is a perfect fit for entry level Human Interface Devices (HID), simple sensor node networks, and remote controlled toys.

The nRF31512 offers 17 kB of one time programmable on-chip program memory.

The devices are supported by the development tools from the nRF24L series from Nordic Semiconductor and an SDK containing examples and a version of the Gazell protocol stack ported for the nRF31 series.

## 1.1 Prerequisites

In order to fully understand the product specification, a good knowledge of electronic and software engineering is necessary.

## 1.2 Writing conventions

This product specification follows a set of typographic rules to make the document consistent and easy to read. The following writing conventions are used:

- Commands, bit state conditions, and register names are written in `Courier New`.
- Pin names and pin signal conditions are written in **Courier New bold**.
- Cross references are *underlined and highlighted in blue*.



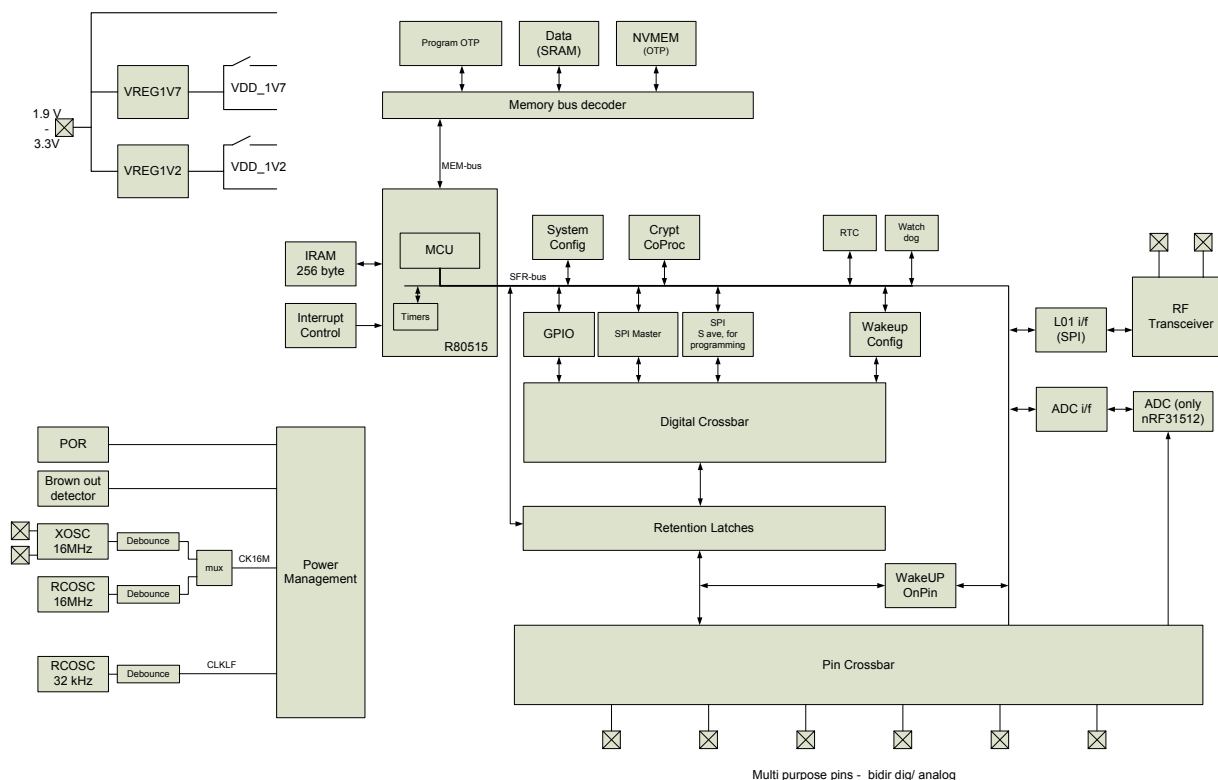
## 2 Product overview

### 2.1 Features

- Fast 8-bit microcontroller:
  - Intel MCS 51 compliant instruction set
  - Reduced instruction cycle time, up to 12 times compared to legacy 8051
- Memory:
  - 17 kB OTP memory for program and data with security features
  - Data memory: 512 bytes of on-chip RAM memory
- A number of on-chip hardware resources are available through programmable multi purpose input/output pins:
  - GPIO
  - SPI master
  - SPI slave for programming and testing
  - ADC (only nRF31512)
  - External interrupts
  - Timer inputs
- High performance 2.4 GHz RF transceiver
  - True single chip GFSK transceiver
  - Enhanced ShockBurst™ link layer support in HW:
    - Packet assembly/disassembly
    - Address and CRC computation
    - Auto ACK and retransmit
  - On the air data rate 250 kbps, 1 Mbps or 2 Mbps
  - 125 RF channels operation, with 79 (2.402 GHz – 2.480 GHz) channels within 2.400 GHz–2.4835 GHz
  - Short switching time enable frequency hopping
  - RF Compatible with nRF24L series
  - RF compatible with nRF2401A, nRF2402, nRF24E1, nRF24E2 in 250 kbps and 1 Mbps mode
- A/D converter (only nRF31512):
  - 8 or 10 bit resolution
  - 2 input channels
  - Single-ended input
  - Full-scale range is three times internal reference voltage
  - Single step mode with conversion time down to 20 µs
  - Mode for measuring supply voltage
- System reset and power supply monitoring:
  - On-chip power-on and brown-out reset
  - Watchdog timer reset
  - Reset from pin
  - Power-fail comparator with programmable threshold and interrupt to MCU
- On-chip timers:
  - Three 16-bit timers/counters operating at the system clock (sources from the 16 MHz on-chip oscillators)
  - One 16-bit timer/counter operating at the low frequency clock (32.768 kHz)
- On-chip oscillators:
  - 16 MHz crystal oscillator XOSC16M
  - 16 MHz RC-oscillator RCOSC16M
  - 32.768 kHz RC-oscillator RCOSC32K

- Power management function:
  - Low power design supporting fully static stop/standby
  - Programmable MCU clock frequency from 125 kHz to 16 MHz
  - On chip voltage regulators supporting low power mode
  - Watchdog and wakeup functionality running in low power mode

## 2.2 Block diagram



**Figure 1** nRF31512 block diagram

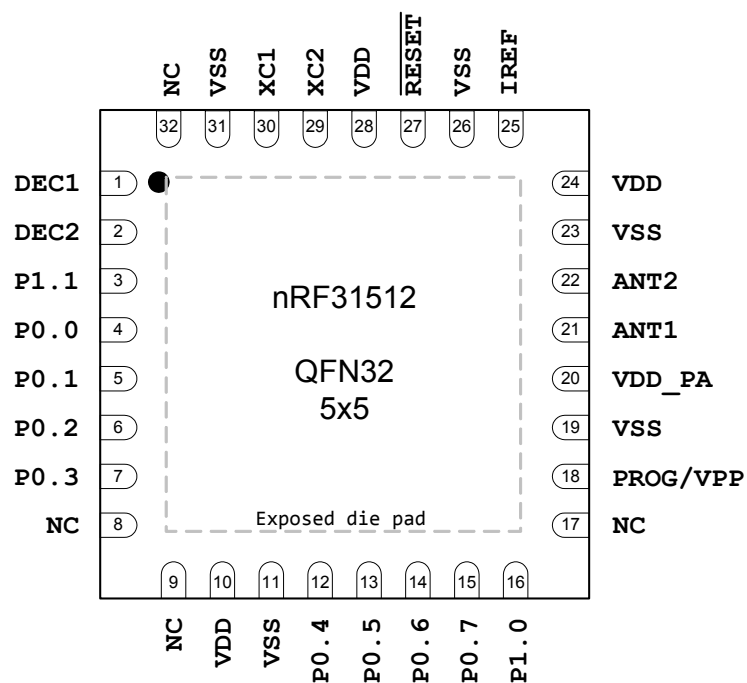
To find more information on the blocks shown in Figure 1, see **Table 1** below:

Name	Reference
Memory (Program, Data, NVMEM)	<i>Chapter 5 on page 63</i>
Power management	<i>Chapter 11 on page 99</i>
RF transceiver	<i>Chapter 3 on page 13</i>
SPI (Master)	<i>Chapter 16 on page 126</i>
GPIO	<i>Chapter 15 on page 117</i>
Watchdog	<i>Chapter 10 on page 97</i>

**Table 1** Block diagram cross references

## 2.3 Pin assignments

### 2.3.1 nRF31512 QFN32 package



**Figure 2** nRF31512 pin assignment (top view) in QFN32 package

## 2.4 Pin functions

Name	Type	Description
<b>VDD</b>	Power	Power supply (+1.9 V to +3.3 V DC)
<b>VSS</b>	Power	Ground (0V)
<b>DEC1</b> <b>DEC2</b>	Power	Power supply outputs for de-coupling purposes (100 nF for DEC1, 33 nF for DEC2)
<b>P0.0 – P0.7</b>	Digital I/O <sup>1</sup>	General purpose I/O pins.
<b>P1.0 – P1.1</b>	Digital I/O	General purpose I/O pins .
<b>PROG/VPP</b>	Digital Input, High Voltage	Input to enable OTP programming on nRF31512. This pin requires an external pull-down resistor, or must be connected to ground if external programming is not needed. For nRF31512 only.
<b>RESET</b>	Digital Input	Reset for microcontroller, active low
<b>IREF</b>	Analog I/O	Device reference current. To be connected to reference resistor on PCB.
<b>VDD_PA</b>	Power Output	Power supply output (+1.8 V) for on-chip RF Power amplifier
<b>ANT1, ANT2</b>	RF	Differential antenna connection (TX and RX)
<b>XC1, XC2</b>	Analog Input	Crystal connection for 16 MHz crystal
Exposed die pad	Power/heat relief	Connect to VSS.

1. P0.4 and P0.5 are ADC inputs in nRF31512

**Table 2** Device pin functions

## 3 RF transceiver

nRF31512 uses the same 2.4 GHz GFSK RF transceiver with embedded protocol engine (Enhanced ShockBurst™) that is found in the nRF24L01+, nRF24LU1+, and nRF24LE1 devices. The RF transceiver is designed for operation in the world wide ISM frequency band at 2.400 GHz–2.4835 GHz and is very well suited for ultra low power wireless applications.

The RF transceiver module is configured and operated through the RF transceiver map. This register map is accessed by the MCU through a dedicated on-chip Serial Peripheral interface (SPI) and is available in all power modes of the RF transceiver module.

The embedded protocol engine (Enhanced ShockBurst™) enables data packet communication and supports various modes from manual operation to advanced autonomous protocol operation. Data FIFOs in the RF transceiver module ensure a smooth data flow between the RF transceiver module and the device's MCU.

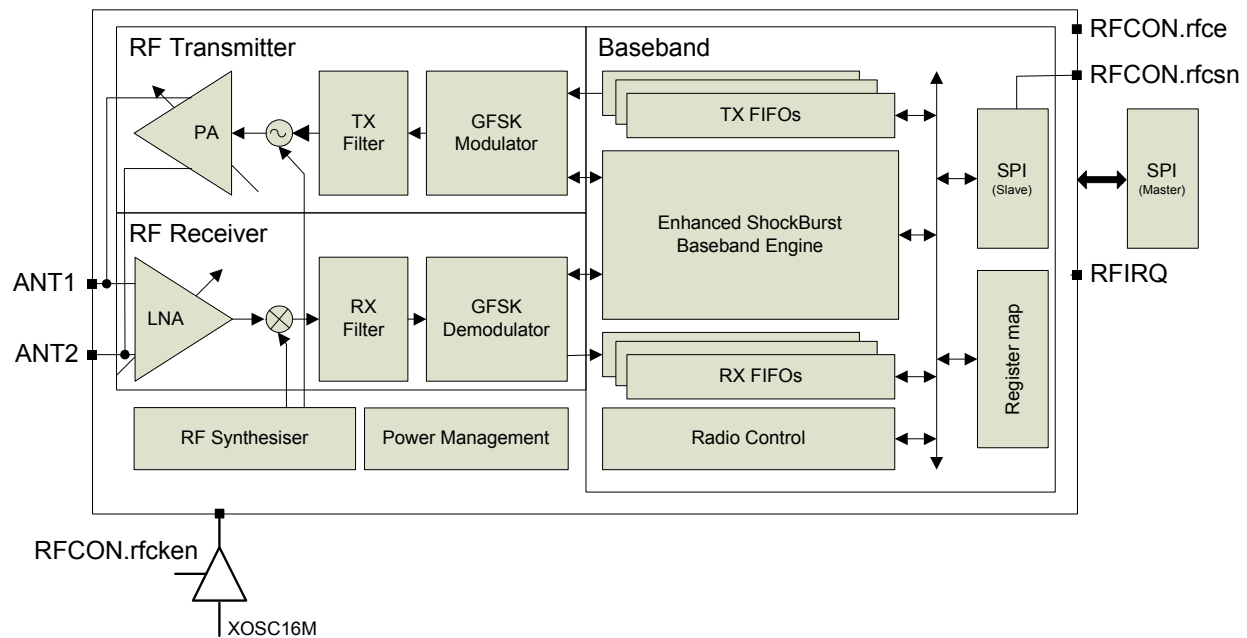
The rest of this chapter is written in the context of the RF transceiver module as the core and the rest of the device as external circuitry to this module.

### 3.1 Features

Features of the RF transceiver include:

- General
  - Worldwide 2.4 GHz ISM band operation
  - Common antenna interface in transmit and receive
  - GFSK modulation
  - 250 kbps, 1 and 2 Mbps on air data rate
- Transmitter
  - Programmable output power: 0, -6, -12 or -18 dBm
  - 11.1mA at 0dBm output power
- Receiver
  - Integrated channel filters
  - 13.3 mA at 2 Mbps
  - -82 dBm sensitivity at 2 Mbps
  - -85 dBm sensitivity at 1 Mbps
  - -94 dBm sensitivity at 250 kbps
- RF Synthesizer
  - Fully integrated synthesizer
  - 1 MHz frequency programming resolution
  - Accepts low cost  $\pm 60$  ppm 16 MHz crystal
  - 1 MHz non-overlapping channel spacing at 1 Mbps
  - 2 MHz non-overlapping channel spacing at 2 Mbps
- Enhanced ShockBurst™
  - 1 to 32 bytes dynamic payload length
  - Automatic packet handling (assembly/disassembly)
  - Automatic packet transaction handling (auto ACK, auto retransmit)
- 2 data pipe MultiCeiver™ for 2:1 star networks

## 3.2 Block diagram



**Figure 3** RF transceiver block diagram

## 3.3 Functional description

This section describes the different operating modes of the RF transceiver and the parameters used to control it.

The RF transceiver module has a built-in state machine that controls the transitions between the different operating modes. The state machine is controlled by SFR register RCON and RF transceiver register CONFIG, see [section 3.5](#) for details.

### 3.3.1 Operational Modes

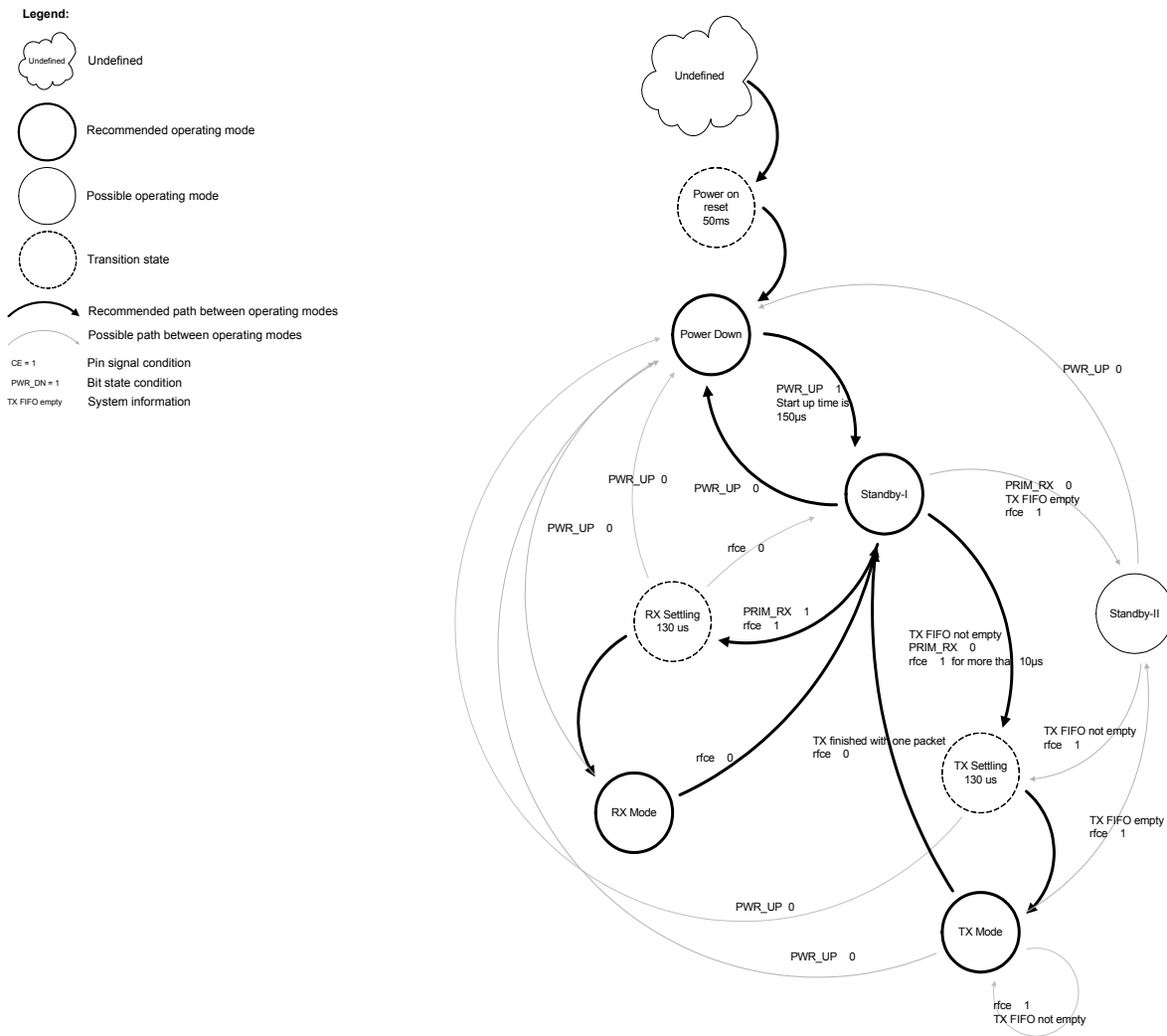
You can configure the RF transceiver to power down, standby, RX and TX mode. This section describes these modes in detail.

### 3.3.1.1 State diagram

The state diagram (**Figure 4**) shows the operating modes of the RF transceiver and how they function. At the end of the reset sequence the RF transceiver enters Power Down mode. When the RF transceiver enters Power Down mode the MCU can still control the module through the SPI and the `r_fcsn` bit in the RCON register.

There are three types of distinct states highlighted in the state diagram:

- **Recommended operating mode:** is a recommended state used during normal operation.
- **Possible operating mode:** is a possible operating state, but is not used during normal operation.
- **Transition state:** is a time limited state used during start up of the oscillator and settling of the PLL.



**Figure 4** Radio control state diagram

### 3.3.1.2 Power down mode

In power down mode the RF transceiver is disabled with minimal current consumption. All the register values available from the SPI are maintained and the SPI can be activated. For start up times see **Table 5 on page 20**. Power down mode is entered by setting the PWR\_UP bit in the CONFIG register low.

### 3.3.1.3 Standby modes

#### Standby-I mode

By setting the PWR\_UP bit in the CONFIG register to 1, the RF transceiver enters standby-I mode. Standby-I mode is used to minimize average current consumption while maintaining short start up times. Change to the active mode only happens if the rfce bit is enabled and when it is not enabled, the RF transceiver returns to standby-I mode from both the TX and RX modes.

#### Standby-II mode

In standby-II mode extra clock buffers are active and more current is used compared to standby-I mode. The RF transceiver enters standby-II mode if the rfce bit is held high on a PTX operation with an empty TX FIFO. If a new packet is downloaded to the TX FIFO, the PLL immediately starts and the packet is transmitted after the normal PLL settling delay (130  $\mu$ s).

The register values are maintained and the SPI can be activated during both standby modes. For start up times see **Table 5 on page 20**.



#### 3.3.1.4 RX mode

The RX mode is an active mode where the RF transceiver is used as a receiver. To enter this mode, the RF transceiver must have the PWR\_UP bit, PRIM\_RX bit and the rfce bit is set high.

In RX mode the receiver demodulates the signals from the RF channel, constantly presenting the demodulated data to the baseband protocol engine. The baseband protocol engine constantly searches for a valid packet. If a valid packet is found (by a matching address and a valid CRC) the payload of the packet is presented in a vacant slot in the RX FIFOs. If the RX FIFOs are full, the received packet is discarded.

The RF transceiver remains in RX mode until the MCU configures it to standby-I mode or power down mode. However, if the automatic protocol features (Enhanced ShockBurst™) in the baseband protocol engine are enabled, the RF transceiver can enter other modes in order to execute the protocol.

In RX mode a Received Power Detector (RPD) signal is available. The RPD is a signal that is set high when a RF signal higher than -64 dBm is detected inside the receiving frequency channel. The internal RPD signal is filtered before presented to the RPD register. The RF signal must be present for at least 40 µs before the RPD is set high. How to use the RPD is described in **Section 3.3.4 on page 20**.

#### 3.3.1.5 TX mode

The TX mode is an active mode for transmitting packets. To enter this mode, the RF transceiver must have the PWR\_UP bit set high, PRIM\_RX bit set low, a payload in the TX FIFO and a high pulse on the rfce bit for more than 10 µs.

The RF transceiver stays in TX mode until it finishes transmitting a packet. If rfce = 0, RF transceiver returns to standby-I mode. If rfce = 1, the status of the TX FIFO determines the next action. If the TX FIFO is not empty the RF transceiver remains in TX mode and transmits the next packet. If the TX FIFO is empty the RF transceiver goes into standby-II mode. The RF transceiver transmitter PLL operates in open loop when in TX mode. It is important never to keep the RF transceiver in TX mode for more than 4 ms at a time. If the Enhanced ShockBurst™ features are enabled, RF transceiver is never in TX mode longer than 4 ms.

### 3.3.1.6 Operational modes configuration

The following table (*Table 3*) describes how to configure the operational modes.

Mode	PWR_UP register	PRIM_RX register	rfce	FIFO state
RX mode	1	1	1	-
TX mode	1	0	1	Data in TX FIFO. Will empty all levels in TX FIFO <sup>1</sup> .
TX mode	1	0	Minimum 10 $\mu$ s high pulse	Data in TX FIFO. Will empty one level in TX FIFO <sup>2</sup> .
Standby-II	1	0	1	TX FIFO empty
Standby-I	1	-	0	No ongoing packet transmission
Power Down	0	-	-	-

1. If the **rfce** bit is held high the TX FIFO is emptied and all necessary ACK and possible retransmits are carried out. The transmission continues as long as the TX FIFO is refilled. If the TX FIFO is empty when the **rfce** bit is still high, the RF transceiver enters standby-II mode. In this mode the transmission of a packet is started as soon as the **rfcsn** is set high after an upload (UL) of a packet to TX FIFO.
2. This operating mode pulses the **rfce** bit high for at least 10  $\mu$ s. This allows one packet to transmit. This is the normal operating mode. After the packet is transmitted, the RF transceiver enters standby-I mode.

*Table 3 RF transceiver main modes*

### 3.3.1.7 Timing information

The timing information in this section relates to the transitions between modes and the timing for the `rfce` bit. The transition from TX mode to RX mode or vice versa is the same as the transition from the standby modes to TX mode or RX mode (130  $\mu$ s), as described in **Table 4**

Name	RF transceiver	Max.	Min.	Comments
Tpd2stby	Power Down → Standby mode	1 $\mu$ s <sup>1</sup>		
Tstby2a	Standby modes → TX/RX mode	130 $\mu$ s		
Thce	Minimum <code>rfce</code> high		10 $\mu$ s	
Tpece2csn	Delay from <code>rfce</code> pos. edge to <code>rfcsn</code> low		4 $\mu$ s	

1. This presupposes that the XO is running. Please refer to CLKLFCTRL for bit 3 in **Table 59 on page 105**.

**Table 4** Operational timing of RF transceiver

**Note:** If `VDD` is turned off, the register values are lost and you must configure the RF transceiver before entering the TX or RX modes.

### 3.3.2 Air data rate

The air data rate is the modulated signaling rate the RF transceiver uses when transmitting and receiving data. It can be 250 kbps, 1 Mbps or 2 Mbps. Using lower air data rate gives better receiver sensitivity than higher air data rate. But, high air data rate gives lower average current consumption and reduced probability of on-air collisions.

The air data rate is set by the `RF_DR` bit in the `RF_SETUP` register. A transmitter and a receiver must be programmed with the same air data rate to communicate with each other.

The RF transceiver is fully compatible with nRF24L01. For compatibility with nRF2401A, nRF2402, nRF24E1, and nRF24E2, the air data rate must be set to 250 kbps or 1 Mbps.

### 3.3.3 RF channel frequency

The RF channel frequency determines the center of the channel used by the RF transceiver. The channel occupies a bandwidth of less than 1 MHz at 250 kbps and 1 Mbps and a bandwidth of less than 2 MHz at 2 Mbps. The RF transceiver can operate on frequencies from 2.400 GHz to 2.525 GHz. The programming resolution of the RF channel frequency setting is 1 MHz.

At 2 Mbps the channel occupies a bandwidth wider than the resolution of the RF channel frequency setting. To ensure non-overlapping channels in 2 Mbps mode, the channel spacing must be 2 MHz or more. At 1 Mbps and 250 kbps the channel bandwidth is the same or lower than the resolution of the RF frequency.

The RF channel frequency is set by the `RF_CH` register according to the following formula:

$$F_0 = 2400 + \text{RF\_CH MHz}$$

You must program a transmitter and a receiver with the same RF channel frequency to communicate with each other.

### 3.3.4 Received Power Detector measurements

Received Power Detector (RPD), located in register 09, bit 0, triggers at received power levels above -64 dBm that are present in the RF channel you receive on. If the received power is less than -64 dBm, RDP = 0.

The RPD can be read out at any time while the RF transceiver is in receive mode. This offers a snapshot of the current received power level in the channel. The RPD is latched whenever a packet is received or when the MCU sets rfce low.

The status of RPD is correct when RX mode is enabled and after a wait time of  $T_{stby2a} + T_{delay\_AGC} = 130 \mu s + 40 \mu s$ . The RX gain varies over temperature which means that the RPD threshold also varies over temperature. The RPD threshold value is reduced by - 5 dB at  $T = -40^{\circ}C$  and increased by + 5 dB at  $85^{\circ}C$ .

### 3.3.5 PA control

The PA (Power Amplifier) control is used to set the output power from the RF transceiver power amplifier. In TX mode PA control has four programmable steps, see **Table 5 on page 20**

The PA control is set by the RF\_PWR bits in the RF\_SETUP register.

SPI RF-SETUP (RF_PWR)	RF output power	DC current consumption
11	0 dBm	11.1 mA
10	-6 dBm	8.8 mA
01	-12 dBm	7.3 mA
00	-18 dBm	6.8 mA

Conditions:  $V_{DD} = 3.0 V$ ,  $V_{SS} = 0 V$ ,  $T_A = 27^{\circ}C$ , Load impedance =  $15 \Omega + j88 \Omega$ .

**Table 5** RF output power setting for the RF transceiver

### 3.3.6 RX/TX control

The RX/TX control is set by PRIM\_RX bit in the CONFIG register and sets the RF transceiver in transmit/ receive.

## 3.4 Enhanced ShockBurst™

Enhanced ShockBurst™ is a packet based data link layer that features automatic packet assembly and timing, automatic acknowledgement and retransmissions of packets. Enhanced ShockBurst™ enables the implementation of ultra low power and high performance communication. The Enhanced ShockBurst™ features enable significant improvements of power efficiency for bi-directional and uni-directional systems, without adding complexity on the host controller side.

### 3.4.1 Features

The main features of Enhanced ShockBurst™ are:

- 1 to 32 bytes dynamic payload length
- Automatic packet handling
- Auto packet transaction handling
  - Auto Acknowledgement
  - Auto retransmit
- 2 data pipe MultiCeiver™ for 1:2 star networks

### 3.4.2 Enhanced ShockBurst™ overview

Enhanced ShockBurst™ uses ShockBurst™ for automatic packet handling and timing. During transmit, ShockBurst™ assembles the packet and clocks the bits in the data packet for transmission. During receive, ShockBurst™ constantly searches for a valid address in the demodulated signal. When ShockBurst™ finds a valid address, it processes the rest of the packet and validates it by CRC. If the packet is valid the payload is moved into a vacant slot in the RX FIFOs. All high speed bit handling and timing is controlled by ShockBurst™.

Enhanced ShockBurst™ features automatic packet transaction handling for the easy implementation of a reliable bi-directional data link. An Enhanced ShockBurst™ packet transaction is a packet exchange between two transceivers, with one transceiver acting as the Primary Receiver (PRX) and the other transceiver acting as the Primary Transmitter (PTX). An Enhanced ShockBurst™ packet transaction is always initiated by a packet transmission from the PTX, the transaction is complete when the PTX has received an acknowledgment packet (ACK packet) from the PRX. The PRX can attach user data to the ACK packet enabling a bi-directional data link.

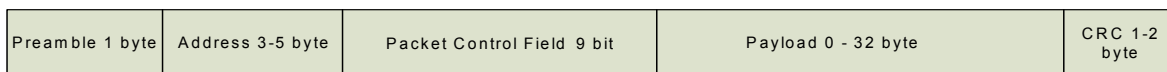
The automatic packet transaction handling works as follows:

1. You begin the transaction by transmitting a data packet from the PTX to the PRX. Enhanced ShockBurst™ automatically sets the PTX in receive mode to wait for the ACK packet.
2. If the packet is received by the PRX, Enhanced ShockBurst™ automatically assembles and transmits an acknowledgment packet (ACK packet) to the PTX before returning to receive mode.
3. If the PTX does not receive the ACK packet immediately, Enhanced ShockBurst™ automatically retransmits the original data packet after a programmable delay and sets the PTX in receive mode to wait for the ACK packet.

In Enhanced ShockBurst™ it is possible to configure parameters such as the maximum number of retransmits and the delay from one transmission to the next retransmission. All automatic handling is done without the involvement of the MCU.

### 3.4.3 Enhanced Shockburst™ packet format

The format of the Enhanced ShockBurst™ packet is described in this section. The Enhanced ShockBurst™ packet contains a preamble field, address field, packet control field, payload field and a CRC field. **Figure 5** shows the packet format with MSB to the left.



**Figure 5** An Enhanced ShockBurst™ packet with payload (0-32 bytes)

#### 3.4.3.1 Preamble

The preamble is a bit sequence used to synchronize the receivers demodulator to the incoming bit stream. The preamble is one byte long and is either 01010101 or 10101010. If the first bit in the address is 1 the preamble is automatically set to 10101010 and if the first bit is 0 the preamble is automatically set to 01010101. This is done to ensure there are enough transitions in the preamble to stabilize the receiver.

#### 3.4.3.2 Address

This is the address for the receiver. An address ensures that the correct packet is detected by the receiver. The address field can be configured to be 3, 4 or, 5 bytes long with the `AW` register.

**Note:** Addresses where the level shifts only one time (that is, 000FFFFFFF) can often be detected in noise and can give a false detection, which may give a raised Packet-Error-Rate. Addresses as a continuation of the preamble (hi-low toggling) raises the Packet-Error-Rate.

#### 3.4.3.3 Packet Control Field (PCF)

**Figure 6** shows the format of the 9 bit packet control field, MSB to the left.



**Figure 6** Packet control field (PCF)

The packet control field contains a 6 bit payload length field, a 2 bit PID (Packet Identity) field and a 1 bit NO\_ACK flag.

#### Payload length

This 6 bit field specifies the length of the payload in bytes. The length of the payload can be from 0 to 32 bytes.

Coding: 000000 = 0 byte (only used in empty ACK packets.) 100000 = 32 byte, 100001 = Don't care.

This field is only used if the Dynamic Payload Length function is enabled.

### PID (Packet identification)

The 2 bit PID field is used to detect if the received packet is new or retransmitted. PID prevents the PRX operation from presenting the same payload more than once to the MCU. The PID field is incremented at the TX side for each new packet received through the SPI. The PID and CRC fields (see **section 3.4.3.5 on page 23**) are used by the PRX operation to determine if a packet is retransmitted or new. When several data packets are lost on the link, the PID fields may become equal to the last received PID. If a packet has the same PID as the previous packet, the RF transceiver compares the CRC sums from both packets. If the CRC sums are also equal, the last received packet is considered a copy of the previously received packet and discarded.

### No Acknowledgment flag (NO\_ACK)

The Selective Auto Acknowledgement feature controls the NO\_ACK flag.

This flag is only used when the auto acknowledgement feature is used. Setting the flag high, tells the receiver that the packet is not to be auto acknowledged.

#### 3.4.3.4 Payload

The payload is the user defined content of the packet. It can be 0 to 32 bytes wide and is transmitted on-air when it is uploaded (unmodified) to the device.

Enhanced ShockBurst™ provides two alternatives for handling payload lengths; static and dynamic.

The default is static payload length. With static payload length all packets between a transmitter and a receiver have the same length. Static payload length is set by the RX\_PW\_Px registers on the receiver side. The payload length on the transmitter side is set by the number of bytes clocked into the TX\_FIFO and must equal the value in the RX\_PW\_Px register on the receiver side.

Dynamic Payload Length (DPL) is an alternative to static payload length. DPL enables the transmitter to send packets with variable payload length to the receiver. This means that for a system with different payload lengths it is not necessary to scale the packet length to the longest payload.

With the DPL feature the device can decode the payload length of the received packet automatically instead of using the RX\_PW\_Px registers. The MCU can read the length of the received payload by using the R\_RX\_PL\_WID command.

**Note:** Always check if the packet width reported is 32 bytes or shorter when using the R\_RX\_PL\_WID command. If its width is longer than 32 bytes then the packet contains errors and must be discarded. Discard the packet by using the Flush\_RX command.

In order to enable DPL the EN\_DPL bit in the FEATURE register must be enabled. In RX mode the DYNPD register must be set. A PTX that transmits to a PRX with DPL enabled must have the DPL\_P0 bit in DYNPD set.

#### 3.4.3.5 CRC (Cyclic Redundancy Check)

The CRC is the error detection mechanism in the packet. It may either be 1 or 2 bytes and is calculated over the address, Packet Control Field and Payload.

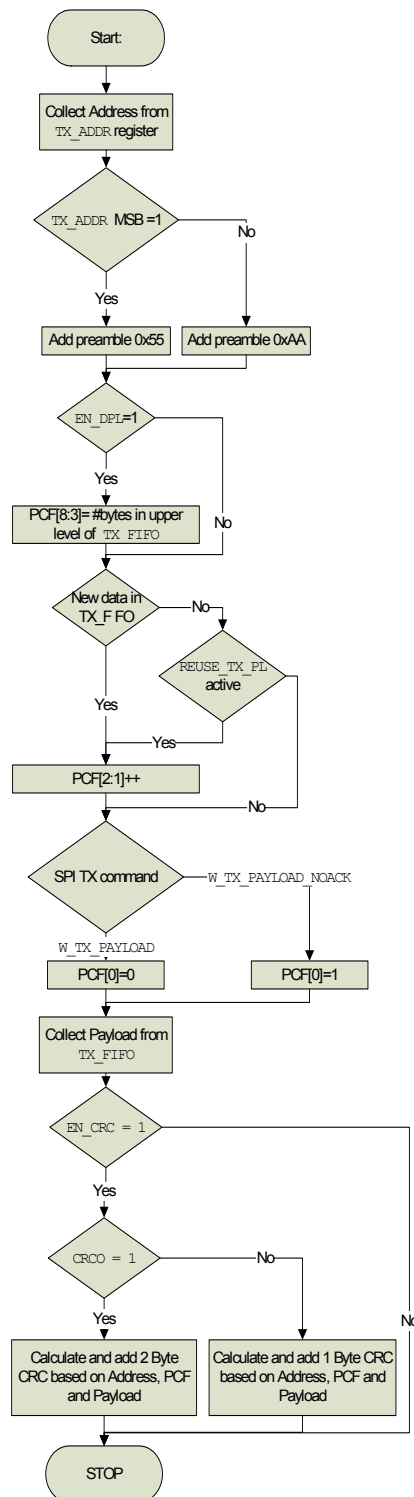
The polynomial for 1 byte CRC is  $X^8 + X^2 + X + 1$ . Initial value 0xFF.

The polynomial for 2 byte CRC is  $X^{16} + X^{12} + X^5 + 1$ . Initial value 0xFFFF.

No packet is accepted by Enhanced ShockBurst™ if the CRC fails.

### 3.4.4 Automatic packet assembly

The automatic packet assembly assembles the preamble, address, packet control field, payload and CRC to make a complete packet before it is transmitted.

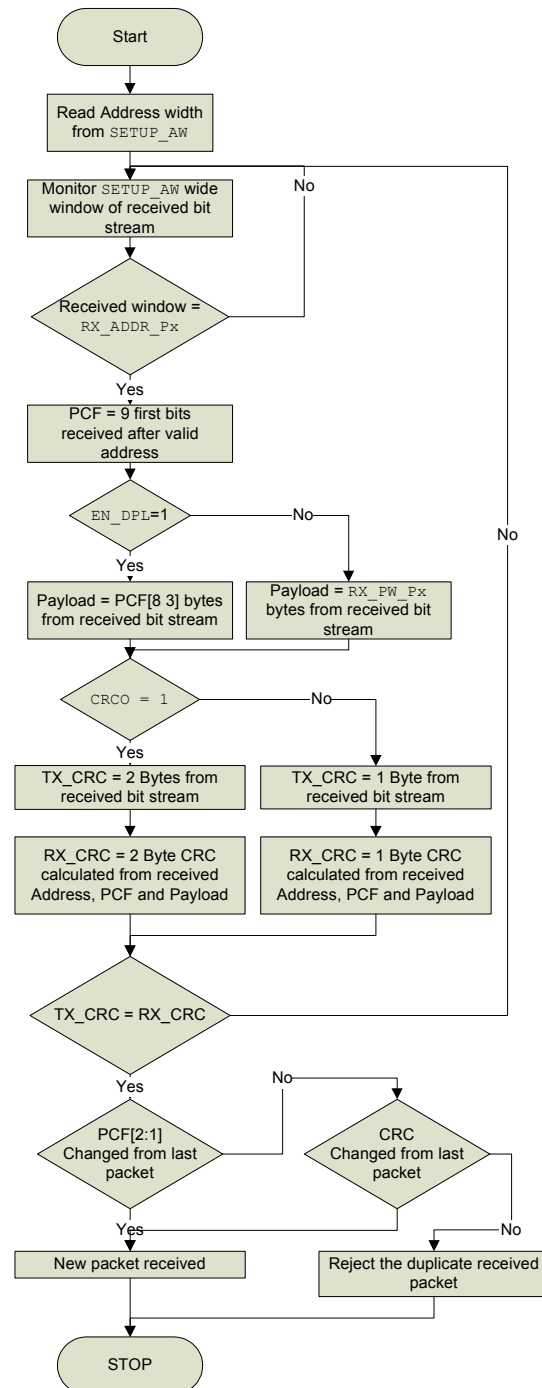


**Figure 7** Automatic packet assembly



### 3.4.5 Automatic packet disassembly

After the packet is validated, Enhanced ShockBurst™ disassembles the packet and loads the payload into the RX FIFO, and asserts the `RX_DR_IRQ`.



**Figure 8** Automatic packet disassembly

### 3.4.6 Automatic packet transaction handling

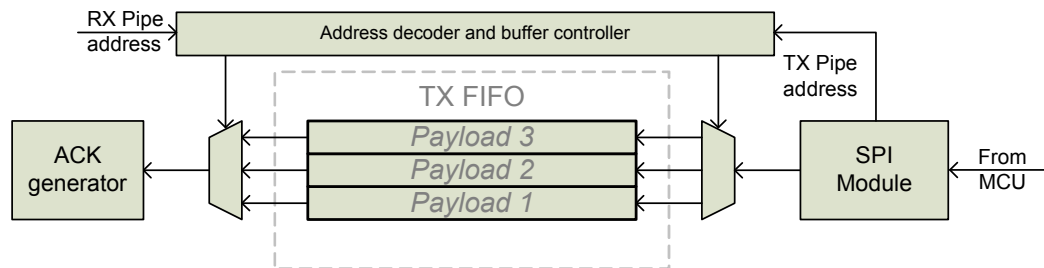
Enhanced ShockBurst™ features two functions for automatic packet transaction handling; auto acknowledgement and auto re-transmit.

#### 3.4.6.1 Auto Acknowledgement

Auto Acknowledgment is a function that automatically transmits an ACK packet to the PTX after it has received and validated a packet. The Auto Acknowledgement function reduces the load of the system MCU and reduces average current consumption. The Auto Acknowledgement feature is enabled by setting the EN\_AA register.

**Note:** If the received packet has the NO\_ACK flag set, auto acknowledgement is not executed.

An ACK packet can contain an optional payload from PRX to PTX. In order to use this feature, the Dynamic Payload Length (DPL) feature must be enabled. The MCU on the PRX side has to upload the payload by clocking it into the TX FIFO by using the W\_ACK\_PAYLOAD command. The payload is pending in the TX FIFO (PRX) until a new packet is received from the PTX. The RF transceiver can have three ACK packet payloads pending in the TX FIFO (PRX) at the same time.



**Figure 9** TX FIFO (PRX) with pending payloads

**Figure 9** shows how the TX FIFO (PRX) is operated when handling pending ACK packet payloads. From the MCU the payload is clocked in with the W\_ACK\_PAYLOAD command. The address decoder and buffer controller ensure that the payload is stored in a vacant slot in the TX FIFO (PRX). When a packet is received, the address decoder and buffer controller are notified with the PTX address. This ensures that the right payload is presented to the ACK generator.

If the TX FIFO (PRX) contains more than one payload to a PTX, payloads are handled using the first in – first out principle. The TX FIFO (PRX) is blocked if all pending payloads are addressed to a PTX where the link is lost. In this case, the MCU can flush the TX FIFO (PRX) by using the FLUSH\_TX command.

In order to enable Auto Acknowledgement with payload the EN\_ACK\_PAY bit in the FEATURE register must be set.

### 3.4.6.2 Auto Retransmission (ART)

The Auto Retransmission is a function that retransmits a packet if an ACK packet is not received. It is used in an Auto Acknowledgement system on the PTX. When a packet is not acknowledged, you can set the number of times it is allowed to retransmit by setting the ARC bits in the SETUP\_RETR register. PTX enters RX mode and waits a time period for an ACK packet each time a packet is transmitted. The amount of time the PTX is in RX mode is based on the following conditions:

- Auto Retransmit Delay (ARD) elapsed.
- No address match within 250  $\mu$ s.
- After received packet (CRC correct or not) if address match within 250  $\mu$ s.

The RF transceiver asserts the TX\_DS IRQ when the ACK packet is received.

The RF transceiver enters standby-I mode if there is no more untransmitted data in the TX FIFO and the rxfce bit in the RFCON register is low. If the ACK packet is not received, the RF transceiver goes back to TX mode after a delay defined by ARD and retransmits the data. This continues until acknowledgment is received, or the maximum number of retransmits is reached.

Two packet loss counters are incremented each time a packet is lost, ARC\_CNT and PLOS\_CNT in the OBSERVE\_TX register. The ARC\_CNT counts the number of retransmissions for the current transaction. You reset ARC\_CNT by initiating a new transaction. The PLOS\_CNT counts the total number of retransmissions since the last channel change. You reset PLOS\_CNT by writing to the RF\_CH register. It is possible to use the information in the OBSERVE\_TX register to make an overall assessment of the channel quality.

The ARD defines the time from the end of a transmitted packet to when a retransmit starts on the PTX. ARD is set in SETUP\_RETR register in steps of 250  $\mu$ s. A retransmit is made if no ACK packet is received by the PTX.

There is a restriction on the length of ARD when using ACK packets with payload. The ARD time must never be shorter than the sum of the startup time and the time on-air for the ACK packet.

- For 2 Mbps data rate and 5-byte address; 15 byte is maximum ACK packet payload length for ARD=250  $\mu$ s (reset value).
- For 1 Mbps data rate and 5-byte address; 5 byte is maximum ACK packet payload length for ARD=250  $\mu$ s (reset value).

ARD=500  $\mu$ s is long enough for any ACK payload length in 1 or 2 Mbps mode.

- For 250 kbps data rate and 5-byte address the following values apply:

ARD	ACK packet size (in bytes)
1500 $\mu$ s	All ACK payload sizes
1250 $\mu$ s	$\leq 24$
1000 $\mu$ s	$\leq 16$
750 $\mu$ s	$\leq 8$
500 $\mu$ s	Empty ACK with no payload

**Table 6** Maximum ACK payload length for different retransmit delays at 250 kbps

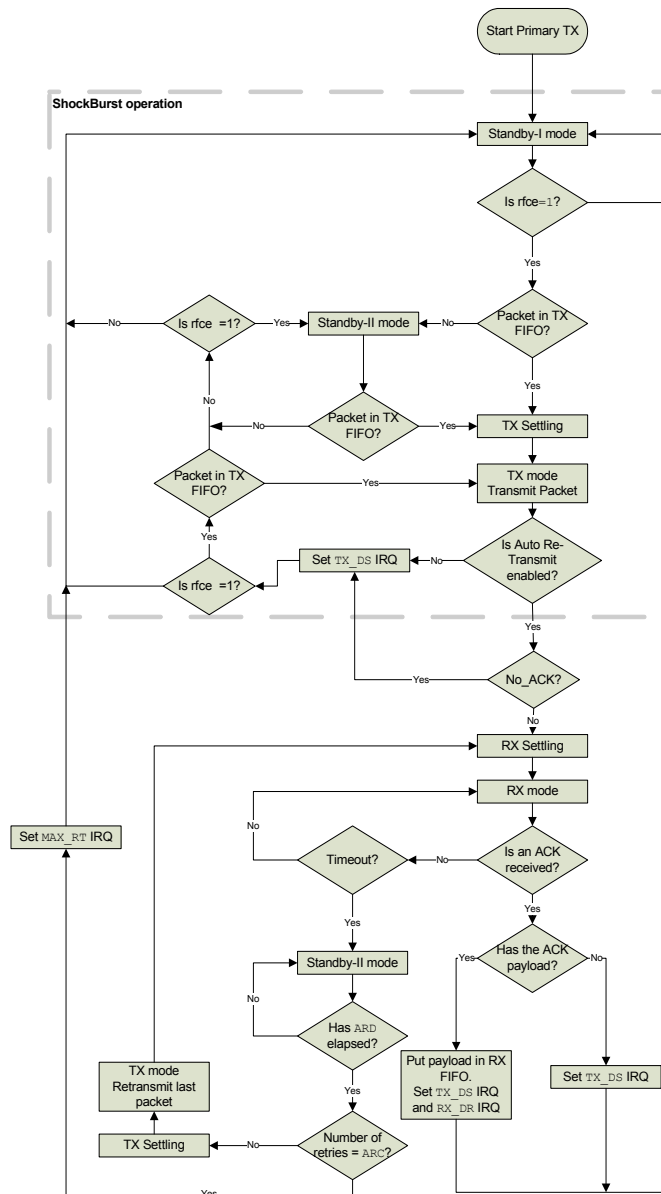
As an alternative to Auto Retransmit it is possible to manually set the RF transceiver to retransmit a packet a number of times. This is done by the REUSE\_TX\_PL command. The MCU must initiate each transmission of the packet with a pulse on the CE pin when this command is used.

### 3.4.7 Enhanced ShockBurst™ flowcharts

This section contains flowcharts outlining PTX and PRX operation in Enhanced ShockBurst™.

#### 3.4.7.1 PTX operation

The flowchart in **Figure 10** outlines how a RF transceiver configured as a PTX behaves after entering standby-I mode.



**Note:** ShockBurst™ operation is outlined with a dashed square.

**Figure 10** PTX operations in Enhanced ShockBurst™

Activate PTX mode by setting the rfce bit in the RCON register high. If there is a packet present in the TX FIFO the RF transceiver enters TX mode and transmits the packet. If Auto Retransmit is enabled, the state

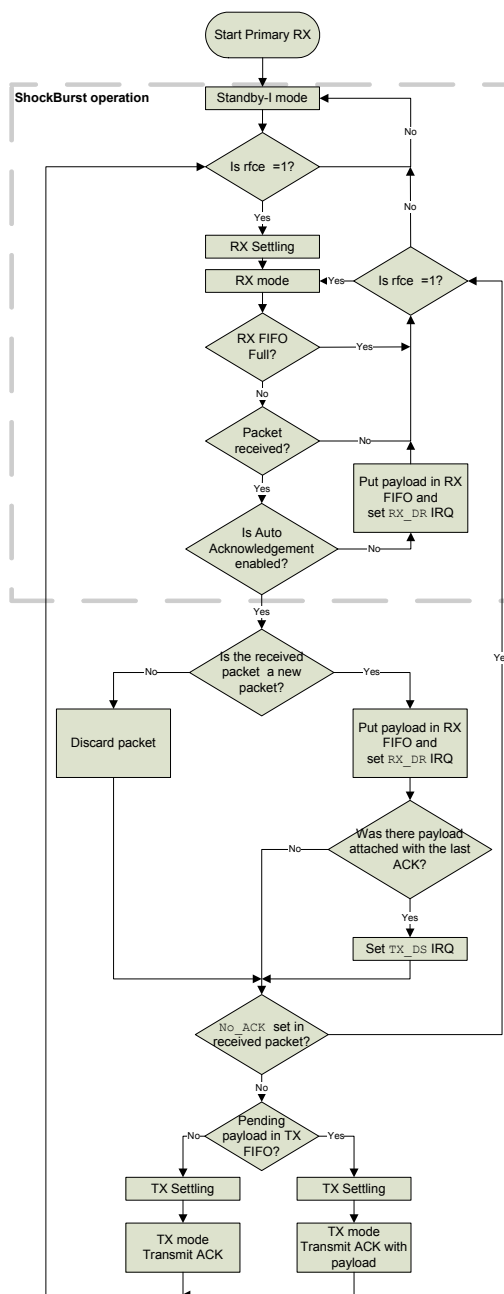
machine checks if the NO\_ACK flag is set. If it is not set, the RF transceiver enters RX mode to receive an ACK packet. If the received ACK packet is empty, only the TX\_DS IRQ is asserted. If the ACK packet contains a payload, both TX\_DS IRQ and RX\_DR IRQ are asserted simultaneously before the RF transceiver returns to standby-I mode.

If the ACK packet is not received before timeout occurs, the RF transceiver returns to standby-II mode. It stays in standby-II mode until the ARD has elapsed. If the number of retransmits has not reached the ARC, the RF transceiver enters TX mode and transmits the last packet once more.

While executing the Auto Retransmit feature, the number of retransmits can reach the maximum number defined in ARC. If this happens, the RF transceiver asserts the MAX\_RT IRQ and returns to standby-I mode.

If the r\_fce bit in the RFCON register is high and the TX FIFO is empty, the RF transceiver enters Standby-II mode.

The flowchart in **Figure 11** outlines how a RF transceiver configured as a PRX behaves after entering standby-I mode.



**Note:** ShockBurst™ operation is outlined with a dashed square.

**Figure 11** PRX operations in Enhanced ShockBurst™

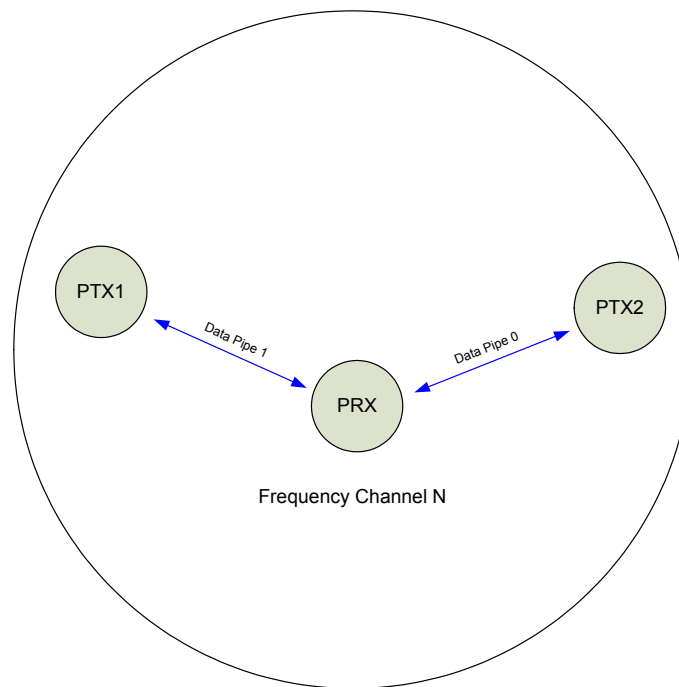
Activate PRX mode by setting the `rfce` bit in the `RFCON` register high. The RF transceiver enters RX mode and starts searching for packets. If a packet is received and Auto Acknowledgement is enabled, the RF transceiver decides if the packet is new or a copy of a previously received packet. If the packet is new the

payload is made available in the RX FIFO and the RX\_DR IRQ is asserted. If the last received packet from the transmitter is acknowledged with an ACK packet with payload, the TX\_DS IRQ indicates that the PTX received the ACK packet with payload. If the No\_ACK flag is not set in the received packet, the PRX enters TX mode. If there is a pending payload in the TX FIFO it is attached to the ACK packet. After the ACK packet is transmitted, the RF transceiver returns to RX mode.

A copy of a previously received packet might be received if the ACK packet is lost. In this case, the PRX discards the received packet and transmits an ACK packet before it returns to RX mode.

### 3.4.8 MultiCeiver™

MultiCeiver™ is a feature used in RX mode that contains a set of two parallel data pipes with unique addresses. A data pipe is a logical channel in the physical RF channel. Each data pipe has its own physical address (data pipe address) decoding in the RF transceiver.



**Figure 12** PRX using MultiCeiver™

The RF transceiver configured as PRX (primary receiver) can receive data addressed to two different data pipes in one frequency channel as shown in **Figure 12**. Each data pipe has its own unique address and can be configured for individual behavior.

Up to two RF transceivers configured as PTX can communicate with one RF transceiver configured as PRX. All data pipe addresses are searched for simultaneously. Only one data pipe can receive a packet at a time. All data pipes can perform Enhanced ShockBurst™ functionality.

The following settings are common to all data pipes:

- CRC enabled/disabled (CRC always enabled when Enhanced ShockBurst™ is enabled)
- CRC encoding scheme
- RX address width
- Frequency channel
- Air data rate
- LNA gain

The data pipes are enabled with the bits in the EN\_RXADDR register. By default only data pipe 0 and 1 are enabled. Each data pipe address is configured in the RX\_ADDR\_PX registers.

**Note:** Always ensure that none of the data pipes have the same address.

Each pipe can have up to a 5 byte configurable address. Data pipe 0 has a unique 5 byte address. The LSByte must be unique for both pipes. **Figure 13** is an example of how data pipes 0-1 are addressed.

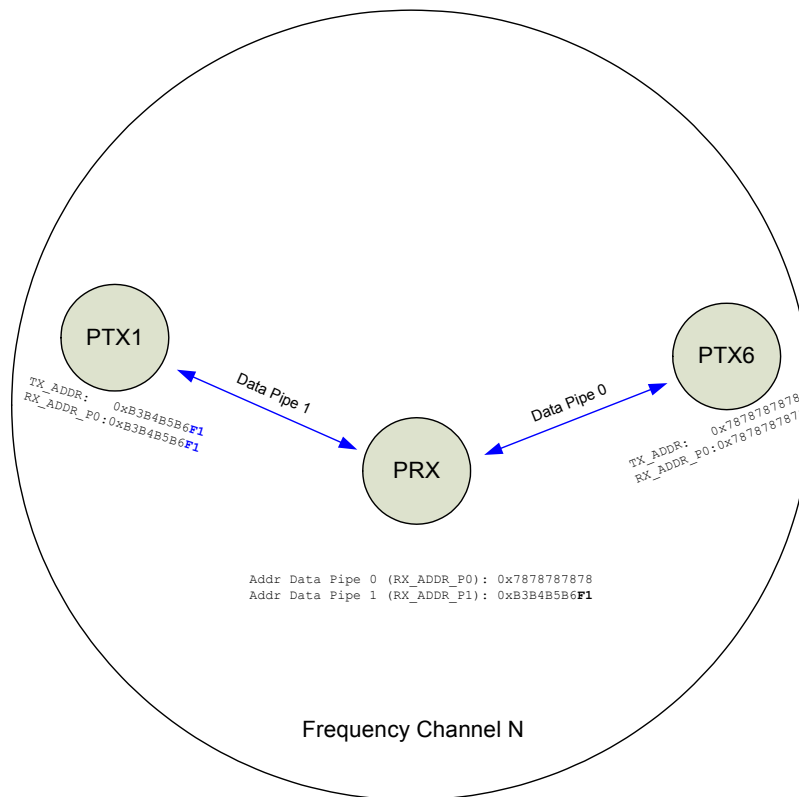
	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
Data pipe 0 (RX_ADDR_P0)	0xE7	0xD3	0xF0	0x35	0x77
Data pipe 1 (RX_ADDR_P1)	0xC2	0xC2	0xC2	0xC2	<b>0xC2</b>

**Figure 13** Addressing data pipes 0-1

The PRX, using MultiCeiver™ and Enhanced ShockBurst™, receives packets from more than one PTX. To ensure that the ACK packet from the PRX is transmitted to the correct PTX, the PRX takes the data pipe address where it received the packet and uses it as the TX address when transmitting the ACK packet. **Figure 14 on page 33** is an example of an address configuration for the PRX and PTX. On the PRX the RX\_ADDR\_Pn,



defined as the pipe address, must be unique. On the PTX the TX\_ADDR must be the same as the RX\_ADDR\_P0 and as the pipe address for the designated pipe.



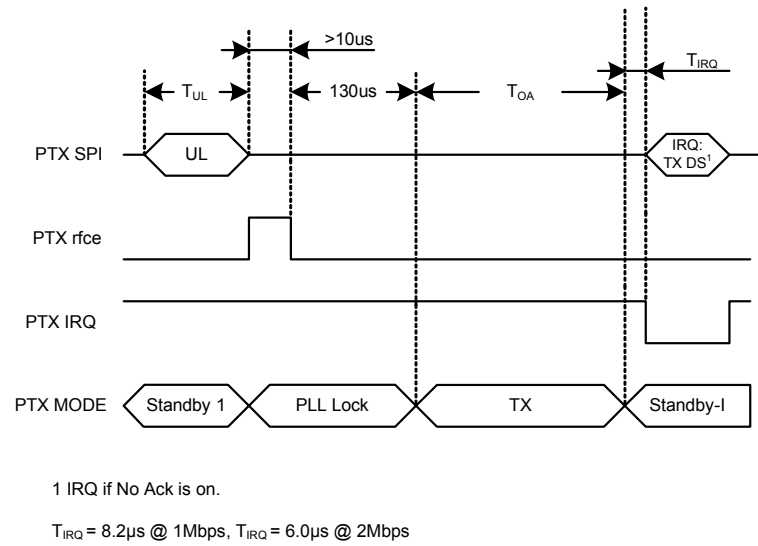
**Figure 14** Example of data pipe addressing in MultiCeiver™

Only when a data pipe receives a complete packet can other data pipes begin to receive data. When multiple PTXs are transmitting to a PRX, the ARD can be used to skew the auto retransmission so that they only block each other once.

### 3.4.9 Enhanced ShockBurst™ timing

This section describes the timing sequence of Enhanced ShockBurst™ and how all modes are initiated and operated. The Enhanced ShockBurst™ timing is controlled through the Data and Control interface. The RF transceiver can be set to static modes or autonomous modes where the internal state machine controls the

events. Each autonomous mode/sequence ends with a RFIRQ interrupt. All the interrupts are indicated as IRQ events in the timing diagrams.

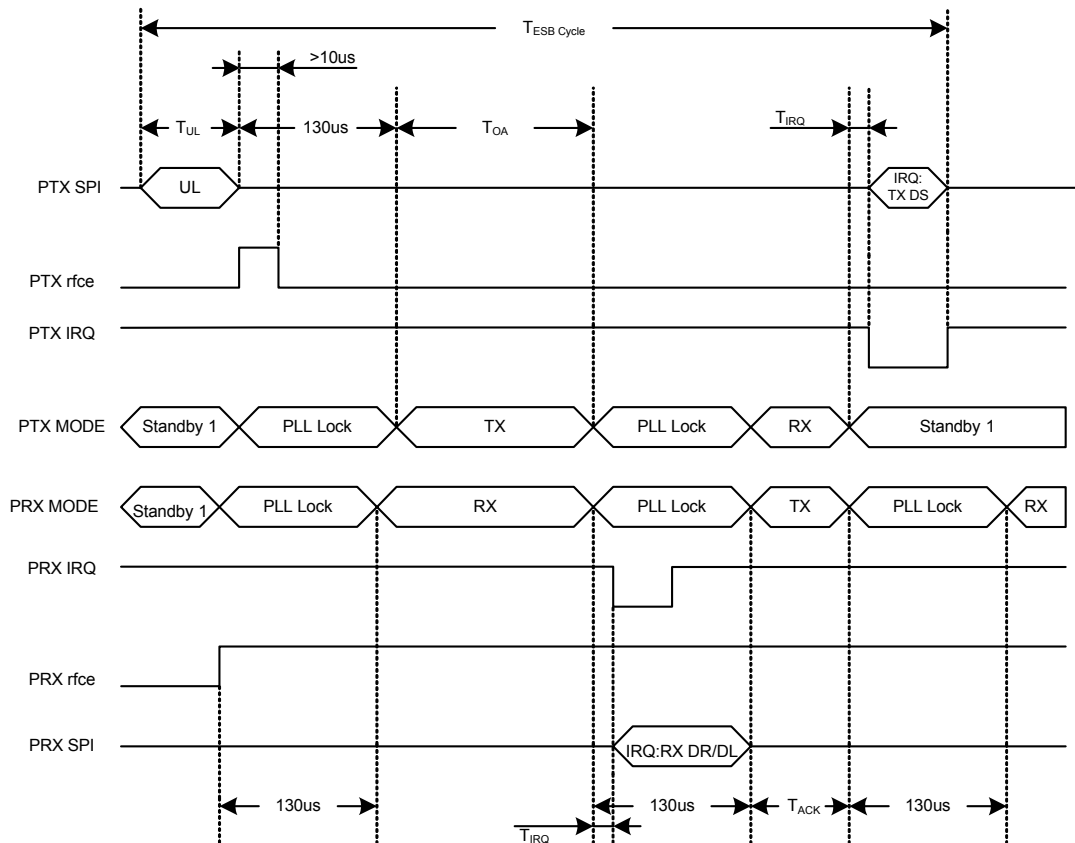


**Figure 15** Transmitting one packet with NO\_ACK on

The following equations calculate various timing measurements:

Symbol	Description	Equation
$T_{\text{OA}}$	Time on-air	$T_{\text{OA}} = \frac{\text{packet length}}{\text{air data rate}} = \frac{8 \left[ \frac{\text{bit}}{\text{byte}} \right] \cdot \left( 1 \left[ \frac{\text{byte}}{\text{preamble}} \right] + 3, 4 \text{ or } 5 \left[ \frac{\text{bytes}}{\text{address}} \right] + N \left[ \frac{\text{bytes}}{\text{payload}} \right] + 1 \text{ or } 2 \left[ \frac{\text{bytes}}{\text{CRC}} \right] \right) + 9 \left[ \frac{\text{bit}}{\text{packet control field}} \right]}{\text{air data rate} \left[ \frac{\text{bit}}{\text{s}} \right]}$
$T_{\text{ACK}}$	Time on-air Ack	$T_{\text{ACK}} = \frac{\text{packet length}}{\text{air data rate}} = \frac{8 \left[ \frac{\text{bit}}{\text{byte}} \right] \cdot \left( 1 \left[ \frac{\text{byte}}{\text{preamble}} \right] + 3, 4 \text{ or } 5 \left[ \frac{\text{bytes}}{\text{address}} \right] + N \left[ \frac{\text{bytes}}{\text{payload}} \right] + 1 \text{ or } 2 \left[ \frac{\text{bytes}}{\text{CRC}} \right] \right) + 9 \left[ \frac{\text{bit}}{\text{packet control field}} \right]}{\text{air data rate} \left[ \frac{\text{bit}}{\text{s}} \right]}$
$T_{\text{UL}}$	Time Upload	$T_{\text{UL}} = \frac{\text{payload length}}{\text{SPI data rate}} = \frac{8 \left[ \frac{\text{bit}}{\text{byte}} \right] \cdot N \left[ \frac{\text{bytes}}{\text{payload}} \right]}{\text{SPI data rate} \left[ \frac{\text{bit}}{\text{s}} \right]}$
$T_{\text{ESB}}$	Time Enhanced ShockBurst™ cycle	$T_{\text{ESB}} = T_{\text{UL}} + 2 \cdot T_{\text{stby2a}} + T_{\text{OA}} + T_{\text{ACK}} + T_{\text{IRQ}}$

**Table 7** Timing equations

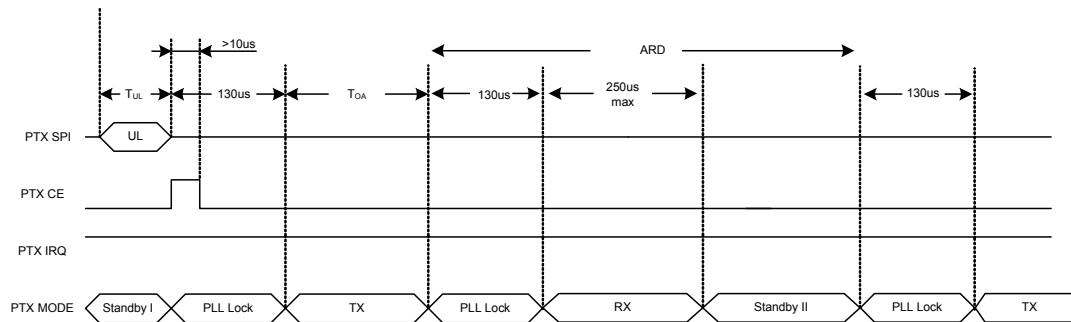


**Figure 16** Timing of Enhanced ShockBurst™ for one packet upload (2 Mbps)

In **Figure 16** the transmission and acknowledgement of a packet is shown. The PRX operation activates RX mode ( $rfce=1$ ), and the PTX operation is activated in TX mode ( $rfce=1$  for minimum 10  $\mu s$ ). After 130  $\mu s$  the transmission starts and finishes after the elapse of  $T_{OA}$ .

When the transmission ends the PTX operation automatically switches to RX mode to wait for the ACK packet from the PRX operation. When the PRX operation receives the packet it sets the interrupt for the host MCU and switches to TX mode to send an ACK. After the PTX operation receives the ACK packet it sets the interrupt to the MCU and clears the packet from the TX FIFO.

In **Figure 17** the PTX timing of a packet transmission is shown when the first ACK packet is lost. To see the complete transmission when the ACK packet fails see **Figure 20 on page 38**.



**Figure 17** Timing of Enhanced ShockBurst™ when the first ACK packet is lost (2 Mbps)

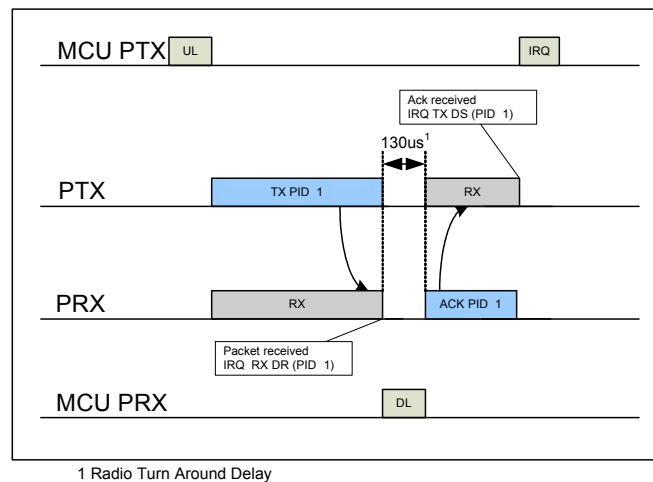
### 3.4.10 Enhanced ShockBurst™ transaction diagram

This section describes several scenarios for the Enhanced ShockBurst™ automatic transaction handling. The call outs in this section's figures indicate the IRQs and other events. For MCU activity the event may be placed at a different timeframe.

**Note:** The figures in this section indicate the earliest possible download (DL) of the packet to the MCU and the latest possible upload (UL) of payload to the transmitter.

### 3.4.10.1 Single transaction with ACK packet and interrupts

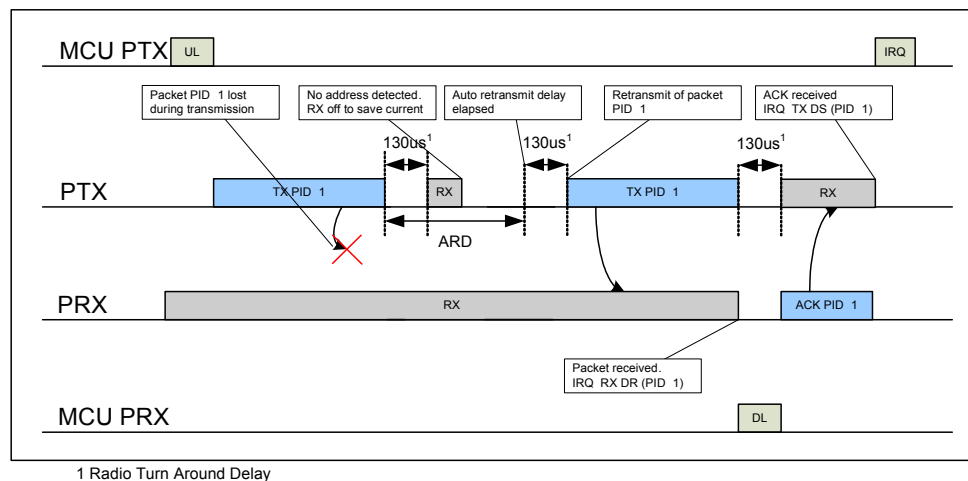
In **Figure 18** the basic auto acknowledgement is shown. After the packet is transmitted by the PTX and received by the PRX the ACK packet is transmitted from the PRX to the PTX. The RX\_DR IRQ is asserted after the packet is received by the PRX, whereas the TX\_DS IRQ is asserted when the packet is acknowledged and the ACK packet is received by the PTX.



**Figure 18** TX/RX cycles with ACK and the according interrupts

### 3.4.10.2 Single transaction with a lost packet

**Figure 19** is a scenario where a retransmission is needed due to loss of the first packet transmit. After the packet is transmitted, the PTX enters RX mode to receive the ACK packet. After the first transmission, the PTX waits a specified time for the ACK packet, if it is not in the specific time slot the PTX retransmits the packet as shown in **Figure 19**

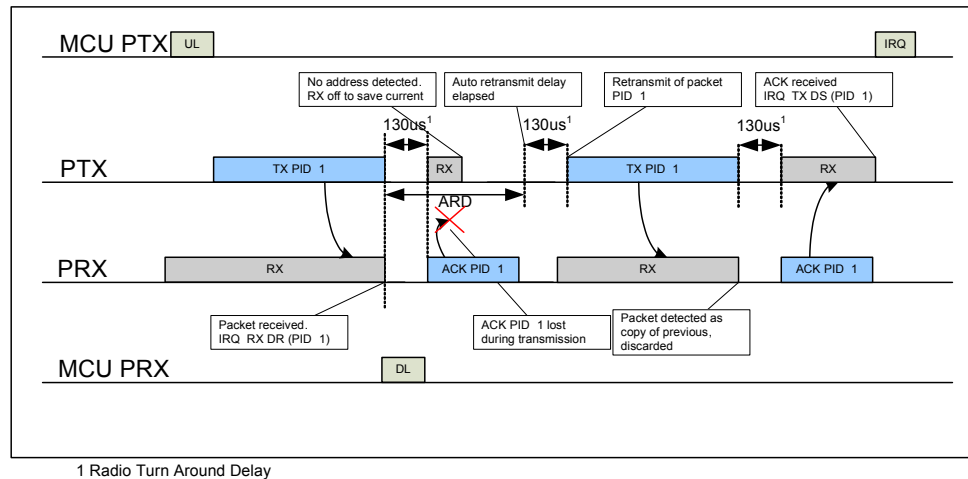


**Figure 19** TX/RX cycles with ACK and the according interrupts when the first packet transmit fails

When an address is detected the PTX stays in RX mode until the packet is received. When the retransmitted packet is received by the PRX (see **Figure 19**), the RX\_DR IRQ is asserted and an ACK is transmitted back to the PTX. When the ACK is received by the PTX, the TX\_DS IRQ is asserted.

### 3.4.10.3 Single transaction with a lost ACK packet

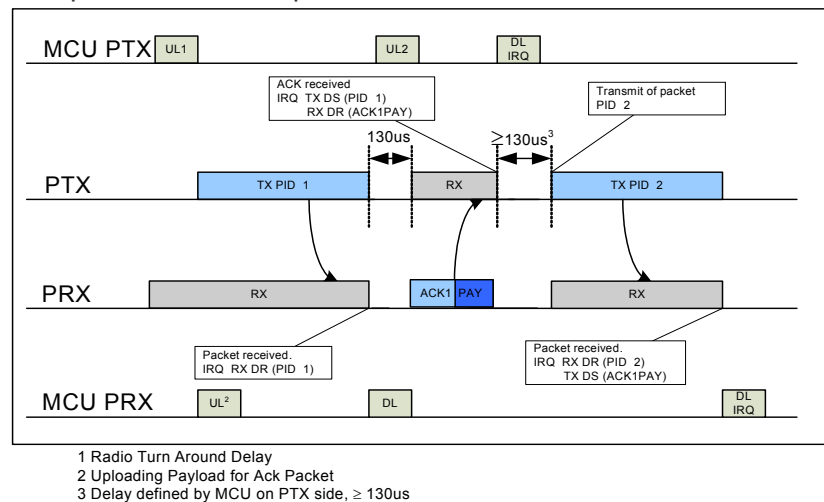
**Figure 20** is a scenario where a retransmission is needed after a loss of the ACK packet. The corresponding interrupts are also indicated.



**Figure 20** TX/RX cycles with ACK and the according interrupts when the ACK packet fails

### 3.4.10.4 Single transaction with ACK payload packet

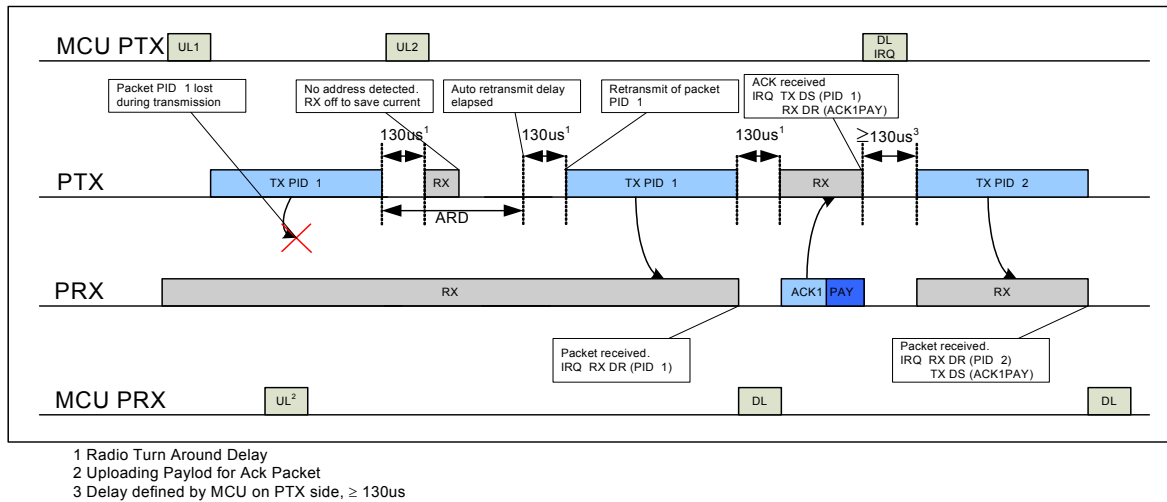
**Figure 21** is a scenario of the basic auto acknowledgement with payload. After the packet is transmitted by the PTX and received by the PRX the ACK packet with payload is transmitted from the PRX to the PTX. The RX\_DR IRQ is asserted after the packet is received by the PRX, whereas on the PTX side the TX\_DS IRQ is asserted when the ACK packet is received by the PTX. On the PRX side, the TX\_DS IRQ for the ACK packet payload is asserted after a new packet from PTX is received. The position of the IRQ in **Figure 21** shows where the MCU can respond to the interrupt.



**Figure 21** TX/RX cycles with ACK Payload and the according interrupts

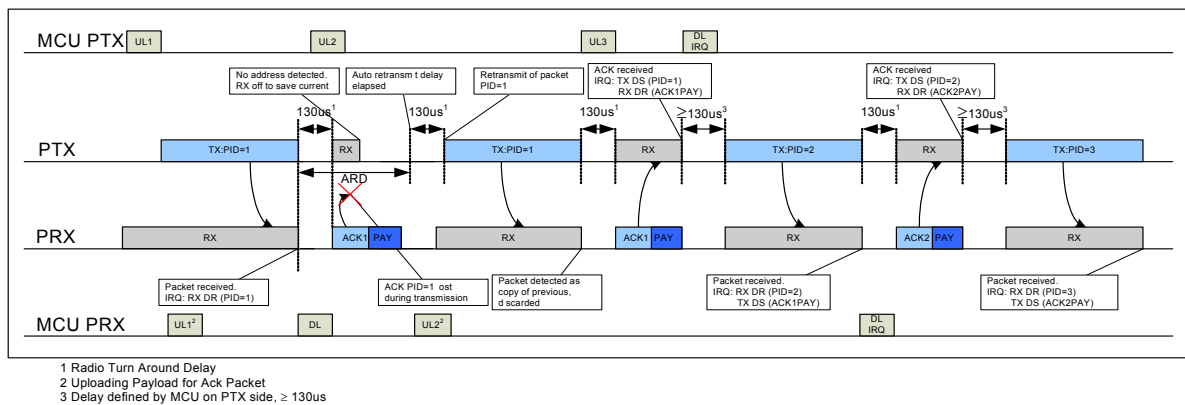
### 3.4.10.5 Single transaction with ACK payload packet and lost packet

**Figure 22** is a scenario where the first packet is lost and a retransmission is needed before the RX\_DR IRQ on the PRX side is asserted. For the PTX both the TX\_DS and RX\_DR IRQ are asserted after the ACK packet is received. After the second packet (PID=2) is received on the PRX side both the RX\_DR (PID=2) and TX\_DS (ACK packet payload) IRQ are asserted.



**Figure 22** TX/RX cycles and the according interrupts when the packet transmission fails

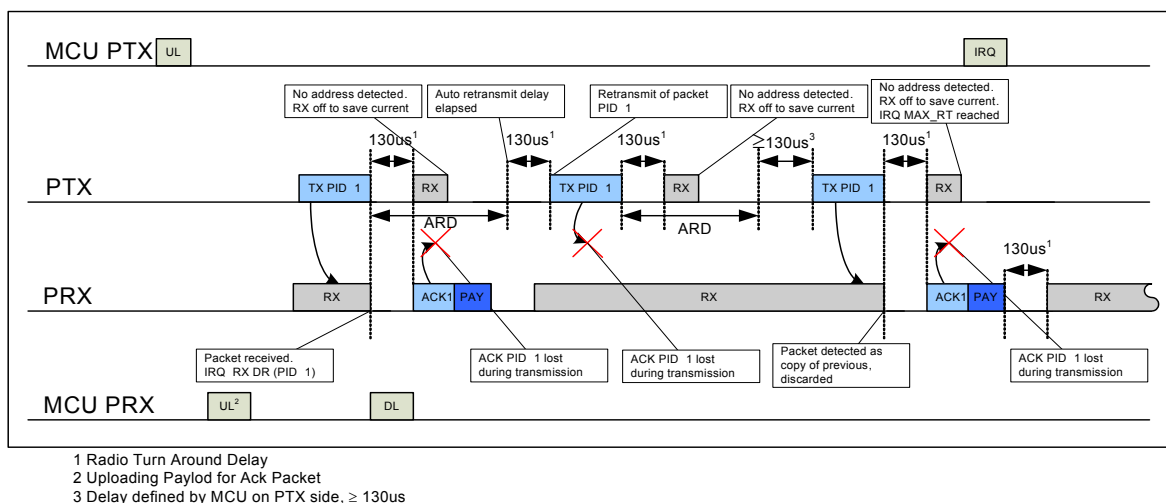
### 3.4.10.6 Two transactions with ACK payload packet and the first ACK packet lost



**Figure 23** TX/RX cycles with ACK Payload and the according interrupts when the ACK packet fails

In **Figure 23** the ACK packet is lost and a retransmission is needed before the TX\_DS IRQ is asserted, but the RX\_DR IRQ is asserted immediately. The retransmission of the packet (PID=1) results in a discarded packet. For the PTX both the TX\_DS and RX\_DR IRQ are asserted after the second transmission of ACK, which is received. After the second packet (PID=2) is received on the PRX both the RX\_DR (PID=2) and TX\_DS (ACK1PAY) IRQ is asserted. The callouts explain the different events and interrupts.

### 3.4.10.7 Two transactions where max retransmissions is reached



**Figure 24** TX/RX cycles with ACK Payload and the according interrupts when the transmission fails. ARC is set to 2.

MAX\_RT IRQ is asserted if the auto retransmit counter (ARC\_CNT) exceeds the programmed maximum limit (ARC). In **Figure 24** the packet transmission ends with a MAX\_RT IRQ. The payload in TX FIFO is NOT removed and the MCU decides the next step in the protocol. A toggle of the rfcfe bit in the RFCON register starts a new transmitting sequence of the same packet. The payload can be removed from the TX FIFO using the FLUSH\_TX command.

### 3.4.11 Compatibility with ShockBurst™

You must disable Enhanced ShockBurst™ for backward compatibility with the nRF2401A, nRF2402, nRF24E1 and, nRF24E2. Set the register EN\_AA = 0x00 and ARC = 0 to disable Enhanced ShockBurst™. In addition, the RF transceiver air data rate must be set to 1 Mbps or 250 kbps.

#### 3.4.11.1 ShockBurst™ packet format

The ShockBurst™ packet format is described in this chapter. **Figure 25** shows the packet format with MSB to the left.

Preamble 1 byte	Address 3-5 byte	Payload 1 - 32 byte	CRC 1-2 byte
-----------------	------------------	---------------------	--------------

**Figure 25** A ShockBurst™ packet compatible with nRF2401/nRF2402/nRF24E1/nRF24E2 devices.

The ShockBurst™ packet format has a preamble, address, payload and CRC field that are the same as the Enhanced ShockBurst™ packet format described in **section 3.4.3 on page 22**.

The differences between the ShockBurst™ packet and the Enhanced ShockBurst™ packet are:

- The 9 bit Packet Control Field is not present in the ShockBurst™ packet format.
- The CRC is optional in the ShockBurst™ packet format and is controlled by the EN\_CRC bit in the CONFIG register.



## 3.5 Data and control interface

The data and control interface gives you access to all the features in the RF transceiver. Compared to the standalone component SFR registers are used instead of port pins. Otherwise the interface is identical to the standalone nRF24L01+ chip.

### 3.5.1 SFR registers

Address (Hex)	Name/Mnemonic	Bit	Reset value	Type	Description
0xE4	SPIRCON0	6:0	0x01	R/W	SPI Master configuration register 0. Reserved. Do not alter.
0xE5	SPIRCON1	3:0	0x0F	R/W	SPI Master configuration register 1.
	<i>maskIrqRxFifoFull</i>	3	1	R/W	1: Disable interrupt when RX FIFO is full. 0: Enable interrupt when RX FIFO is full.
	<i>maskIrqRxDataReady</i>	2	1	R/W	1: Disable interrupt when data is available in RX FIFO. 0: Enable interrupt when data is available in RX FIFO.
	<i>maskIrqTxFifoEmpty</i>	1	1	R/W	1: Disable interrupt when TX FIFO is empty. 0: Enable interrupt when TX FIFO is empty.
	<i>maskIrqTxFifoReady</i>	0	1	R/W	1: Disable interrupt when a location is available in TX FIFO. 0: Enable interrupt when a location is available in TX FIFO.
0xE6	spiMasterStatus SPIRSTAT	3:0	0x03	R	SPI Master status register.
	<i>rxFifoFull</i>	3	0	R	Interrupt source. 1: RX FIFO full. 0: RX FIFO can accept more data from SPI. Cleared when the cause is removed.
	<i>rxDataReady</i>	2	0	R	Interrupt source. 1: Data available in RX FIFO. 0: No data in RX FIFO. Cleared when the cause is removed.
	<i>txFifoEmpty</i>	1	1	R	Interrupt source. 1: TX FIFO empty. 0: Data in TX FIFO. Cleared when the cause is removed.
	<i>txFifoReady</i>	0	1	R	Interrupt source. 1: Location available in TX FIFO. 0: TX FIFO full. Cleared when the cause is removed.
0xE7	SPIRDAT	7:0	0x00	R/W	SPI Master data register. Accesses TX (write) and RX (read) FIFO buffers, both two bytes deep.

**Table 8** RF transceiver SPI master registers

The RF transceiver SPI Master is configured through SPIRCON1. Four different sources can generate interrupt, unless they are masked by their respective bits in SPIRCON1. SPIRSTAT reveals which sources that are active.

SPIRDAT accesses both the TX (write) and the RX (read) FIFOs, which are two bytes deep. The FIFOs are dynamic and can be refilled according to the state of the status flags: "FIFO ready" means that the FIFO can accept data. "Data ready" means that the FIFO can provide data, minimum one byte.

Addr	Bit	Name	R/W	Function
0xE8	7:3	-		Reserved
	2	rfcken	RW	RF Clock Enable (16 MHz)
	1	rfcsn	RW	Enable RF command. 0: enabled
	0	rfce	RW	Enable RF transceiver. 1: enabled

**Table 9** RFCON register

RFCON controls the RF transceiver SPI Slave chip select signal (CSN), the RF transceiver chip enable signal (CE) and the RF transceiver clock enable signal (CKEN).

### 3.5.2 SPI operation

This section describes the SPI commands and timing.

#### 3.5.2.1 SPI commands

The SPI commands are shown in **Table 10 on page 43** Every new command must be started by writing 0 to rfcsn in the RFCON register.

The SPI command is transferred to RF transceiver by writing the command to the SPIRDAT register. After the first transfer the RF transceiver's STATUS register can be read from SPIRDAT when the transfer is completed.

The serial shifting SPI commands is in the following format:

<**Command word:** MSBit to LSBit (one byte)>

<**Data bytes:** LSByte to MSByte, MSBit in each byte first>

.

Command name	Command word (binary)	# Data bytes	Operation
R_REGISTER	000A AAAA	1 to 5 LSByte first	Read command and <code>status</code> registers. AAAAA = 5 bit Register Map Address
W_REGISTER	001A AAAA	1 to 5 LSByte first	Write command and status registers. AAAAA = 5 bit Register Map Address Executable in power down or standby modes only.
R_RX_PAYLOAD	0110 0001	1 to 32 LSByte first	Read RX-payload: 1 – 32 bytes. A read operation always starts at byte 0. Payload is deleted from FIFO after it is read. Used in RX mode.
W_TX_PAYLOAD	1010 0000	1 to 32 LSByte first	Write TX-payload: 1 – 32 bytes. A write operation always starts at byte 0 used in TX payload.
FLUSH_TX	1110 0001	0	Flush TX FIFO, used in TX mode
FLUSH_RX	1110 0010	0	Flush RX FIFO, used in RX mode Should not be executed during transmission of acknowledge, that is, acknowledge package will not be completed.
REUSE_TX_PL	1110 0011	0	Used for a PTX operation Reuse last transmitted payload. TX payload reuse is active until W_TX_PAYLOAD or FLUSH TX is executed. TX payload reuse must not be activated or deactivated during package transmission.
R_RX_PL_WID <sup>1</sup>	0110 0000	1	Read RX payload width for the top R_RX_PAYLOAD in the RX FIFO.  <b>Note:</b> Flush RX FIFO if the read value is larger than 32 bytes.
W_ACK_PAYLOAD <sup>1</sup>	1010 1PPP	1 to 32 LSByte first	Used in RX mode. Write Payload to be transmitted together with ACK packet on PIPE PPP. (PPP valid in the range from 000 to 001). Maximum three ACK packet payloads can be pending. Payloads with same PPP are handled using first in - first out principle. Write payload: 1– 32 bytes. A write operation always starts at byte 0.
W_TX_PAYLOAD_NOACK <sup>1</sup>	1011 0000	1 to 32 LSByte first	Used in TX mode. Disables AUTOACK on this specific packet.
NOP	1111 1111	0	No Operation. Might be used to read the STATUS register

1. The bits in the FEATURE register shown in Table 11 on page 51 have to be set.

**Table 10** Command set for the RF transceiver SPI

The W\_REGISTER and R\_REGISTER commands operate on single or multi-byte registers. When accessing multi-byte registers read or write to the MSBit of LSByte first. You can terminate the writing before all bytes in a multi-byte register are written, leaving the unwritten MSByte(s) unchanged. For example, the LSByte of RX\_ADDR\_P0 can be modified by writing only one byte to the RX\_ADDR\_P0 register. The content of the `status` register is always read to **MISO** after a high to low transition on **CSN**.

**Note:** The 3 bit pipe information in the `STATUS` register is updated during the RFIRQ high to low transition. The pipe information is unreliable if the `STATUS` register is read during an RFIRQ high to low transition.

### 3.5.3 Data FIFO

The data FIFOs store transmitted payloads (TX FIFO) or received payloads that are ready to be clocked out (RX FIFO). The FIFOs are accessible in both PTX mode and PRX mode.

The following FIFOs are present in the RF transceiver:

- TX three level, 32 byte FIFO
- RX three level, 32 byte FIFO

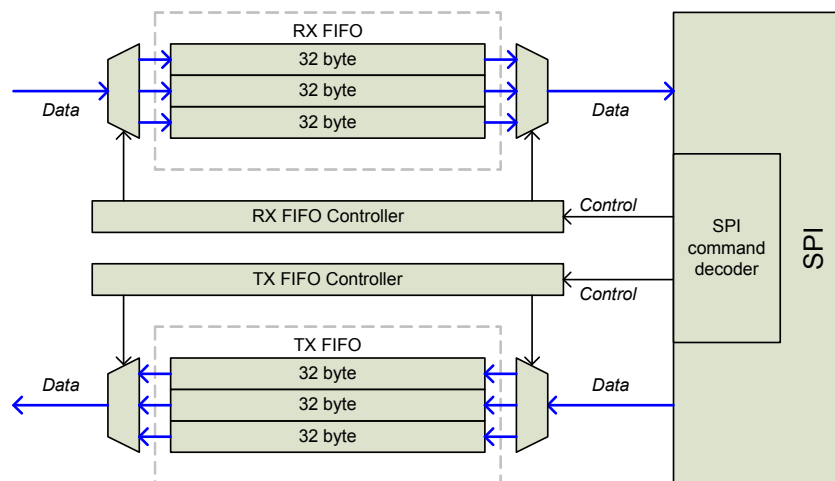
Both FIFOs have a controller and are accessible through the SPI by using dedicated SPI commands. A TX FIFO in PRX can store payloads for ACK packets to three different PTX operations. If the TX FIFO contains more than one payload to a pipe, payloads are handled using the first in - first out principle. The TX FIFO in a PRX is blocked if all pending payloads are addressed to pipes where the link to the PTX is lost. In this case, the MCU can flush the TX FIFO using the FLUSH\_TX command.

The RX FIFO in PRX can contain payloads from up to three different PTX operations and a TX FIFO in PTX can have up to three payloads stored.

You can write to the TX FIFO using these three commands; W\_TX\_PAYLOAD and W\_TX\_PAYLOAD\_NO\_ACK in PTX mode and W\_ACK\_PAYLOAD in PRX mode. All three commands provide access to the TX\_PLD register.

The RX FIFO can be read by the command R\_RX\_PAYLOAD in PTX and PRX mode. This command provides access to the RX\_PLD register.

The payload in TX FIFO in a PTX is not removed if the MAX\_RT IRQ is asserted.



**Figure 26** FIFO (RX and TX) block diagram

You can read if the TX and RX FIFO are full or empty in the FIFO\_STATUS register. TX\_REUSE (also available in the FIFO\_STATUS register) is set by the SPI command REUSE\_TX\_PL, and is reset by the SPI commands W\_TX\_PAYLOAD or FLUSH\_TX.

### 3.5.4 Interrupt

The RF transceiver can send interrupts to the MCU. The interrupt (RFIRQ) is activated when TX\_DS, RX\_DR or MAX\_RT are set high by the state machine in the STATUS register. RFIRQ is deactivated when the MCU writes '1' to the interrupt source bit in the STATUS register. The interrupt mask in the CONFIG register is

used to select the IRQ sources that are allowed to activate RFIRQ. By setting one of the mask bits high, the corresponding interrupt source is disabled. By default all interrupt sources are enabled.

**Note:** The 3 bit pipe information in the `STATUS` register is updated during the RFIRQ high to low transition. The pipe information is unreliable if the `STATUS` register is read during a RFIRQ high to low transition.

## 3.6 Register map

You can configure and control the radio (using read and write commands) by accessing the register map through the SPI.

### 3.6.1 Register map table

All undefined bits in the table below are redundant. They are read out as '0'.

**Note:** Addresses 18 to 1B are reserved for test purposes, altering them makes the chip malfunction.

Address (Hex)	Mnemonic	Bit	Reset Value	Type	Description
00	CONFIG				Configuration Register
	Reserved	7	0	R/W	Only '0' allowed
	MASK_RX_DR	6	0	R/W	Mask interrupt caused by RX_DR 1: Interrupt not reflected on the RFIRQ 0: Reflect RX_DR as active low on RFIRQ
	MASK_TX_DS	5	0	R/W	Mask interrupt caused by TX_DS 1: Interrupt not reflected on the RFIRQ 0: Reflect TX_DS as active low interrupt on RFIRQ
	MASK_MAX_RT	4	0	R/W	Mask interrupt caused by MAX_RT 1: Interrupt not reflected on RFIRQ 0: Reflect MAX_RT as active low on RFIRQ
	EN_CRC	3	1	R/W	Enable CRC. Forced high if one of the bits in the EN_AA is high
	CRCO	2	0	R/W	CRC encoding scheme '0' - 1 byte '1' - 2 bytes
	PWR_UP	1	0	R/W	1: POWER UP, 0: POWER DOWN
	PRIM_RX	0	0	R/W	RX/TX control 1: PRX, 0: PTX
01	EN_AA Enhanced ShockBurst™				Enable 'Auto Acknowledgment' Function Disable this functionality to be compatible with nRF2401.
	Reserved	7:2	0	R/W	Only '0' allowed
	ENAA_P1	1	1	R/W	Enable auto acknowledgement data pipe 1
	ENAA_P0	0	1	R/W	Enable auto acknowledgement data pipe 0
02	EN_RXADDR				Enabled RX Addresses
	Reserved	7:2	0	R/W	Only '0' allowed
	ERX_P1	1	1	R/W	Enable data pipe 1.
	ERX_P0	0	1	R/W	Enable data pipe 0.

Address (Hex)	Mnemonic	Bit	Reset Value	Type	Description
03	SETUP_AW				Setup of Address Widths (common for all data pipes)
	Reserved	7:2	000000	R/W	Only '000000' allowed
	AW	1:0	11	R/W	RX/TX Address field width '00' - Illegal '01' - 3 bytes '10' - 4 bytes '11' - 5 bytes LSByte is used if address width is below 5 bytes
04	SETUP_RETR				Setup of Automatic Retransmission
	ARD <sup>1</sup>	7:4	0000	R/W	Auto Retransmit Delay '0000' – Wait 250 µs '0001' – Wait 500 µs '0010' – Wait 750 µs ..... '1111' – Wait 4000 µs (Delay defined from end of transmission to start of next transmission) <sup>2</sup>
	ARC	3:0	0011	R/W	Auto Retransmit Count '0000' – Re-Transmit disabled '0001' – Up to 1 Re-Transmit on fail of AA ..... '1111' – Up to 15 Re-Transmit on fail of AA
05	RF_CH				RF Channel
	Reserved	7	0	R/W	Only '0' allowed
	RF_CH	6:0	0000010	R/W	Sets the frequency channel the RF transceiver operates on
06	RF_SETUP				RF Setup Register
	CONT_WAVE	7	0	R/W	Enables continuous carrier transmit when high.
	Reserved	6	0	R/W	Only '0' allowed
	RF_DR_LOW	5	0	R/W	Set RF Data Rate to 250 kbps. See RF_DR_HIGH for encoding.
	PLL_LOCK	4	0	R/W	Force PLL lock signal. Only used in test
	RF_DR_HIGH	3	1	R/W	Select between the high speed data rates. This bit is don't care if RF_DR_LOW is set. Encoding: RF_DR_LOW, RF_DR_HIGH: '00' – 1 Mbps '01' – 2 Mbps '10' – 250 kbps '11' – Reserved

Address (Hex)	Mnemonic	Bit	Reset Value	Type	Description
	<i>RF_PWR</i>	2:1	11	R/W	Set RF output power in TX mode '00' – -18 dBm '01' – -12 dBm '10' – -6 dBm '11' – 0 dBm
	obsolete	0			Don't care
07	STATUS				Status Register (In parallel to the SPI command word applied on the <b>MOSI</b> pin, the <b>STATUS</b> register is shifted serially out on the <b>MISO</b> pin)
	Reserved	7	0	R/W	Only '0' allowed
	<i>RX_DR</i>	6	0	R/W	Data Ready RX FIFO interrupt. Asserted when new data arrives RX FIFO <sup>3</sup> . Write 1 to clear bit.
	<i>TX_DS</i>	5	0	R/W	Data Sent TX FIFO interrupt. Asserted when packet transmitted on TX. If <b>AUTO_ACK</b> is activated, this bit is set high only when ACK is received. Write 1 to clear bit.
	<i>MAX_RT</i>	4	0	R/W	Maximum number of TX retransmits interrupt Write 1 to clear bit. If <b>MAX_RT</b> is asserted it must be cleared to enable further communication.
	<i>RX_P_NO</i>	3:1	111	R	Data pipe number for the payload available for reading from <b>RX_FIFO</b> 000-001: Data Pipe Number 010-110: Not Used 111: RX FIFO Empty
	<i>TX_FULL</i>	0	0	R	TX FIFO full flag. 1: TX FIFO full. 0: Available locations in TX FIFO.
08	OBSERVE_TX				Transmit observe register
	<i>PLOS_CNT</i>	7:4	0	R	Count lost packets. The counter is overflow protected to 15, and discontinues at max until reset. The counter is reset by writing to <b>RF_CH</b> .
	<i>ARC_CNT</i>	3:0	0	R	Count retransmitted packets. The counter is reset when transmission of a new packet starts.
09	RPD				
	Reserved	7:1	000000	R	
	<i>RPD</i>	0	0	R	Received Power Detector. This register is called CD (Carrier Detect) in the nRF24L01. The name is different in the RF transceiver due to the different input power level threshold for this bit. See <a href="#">section 3.3.4 on page 20</a> .
0A	RX_ADDR_P0	39:0	0xE7E7E7E7	R/W	Receive address data pipe 0. 5 Bytes maximum length. (LSByte is written first. Write the number of bytes defined by <b>SETUP_AW</b> )



Address (Hex)	Mnemonic	Bit	Reset Value	Type	Description
0B	RX_ADDR_P1	39:0	0xC2C2C2C2C2	R/W	Receive address data pipe 1. 5 Bytes maximum length. (LSByte is written first. Write the number of bytes defined by SETUP_AW)
0C	Reserved	7:0	0xC3	R/W	Not used
0D	Reserved	7:0	0xC4	R/W	Not used
0E	Reserved	7:0	0xC5	R/W	Not used
0F	Reserved	7:0	0xC6	R/W	Not used
10	TX_ADDR	39:0	0xE7E7E7E7E7	R/W	Transmit address. Used for a PTX operation only. (LSByte is written first) Set RX_ADDR_P0 equal to this address to handle automatic acknowledge if this is a PTX operation with Enhanced ShockBurst™ enabled.
11	RX_PW_P0				
	Reserved	7:6	00	R/W	Only '00' allowed
	RX_PW_P0	5:0	0	R/W	Number of bytes in RX payload in data pipe 0 (1 to 32 bytes). 0 Pipe not used 1 = 1 byte ... 32 = 32 bytes
12	RX_PW_P1				
	Reserved	7:6	00	R/W	Only '00' allowed
	RX_PW_P1	5:0	0	R/W	Number of bytes in RX payload in data pipe 1 (1 to 32 bytes). 0 Pipe not used 1 = 1 byte ... 32 = 32 bytes
13	RX_PW_P2				
	Reserved	7:0	0	R/W	Only '0' allowed
14	RX_PW_P3				
	Reserved	7:0	0	R/W	Only '0' allowed
15	RX_PW_P4				
	Reserved	7:0	0	R/W	Only '0' allowed
16	RX_PW_P5				
	Reserved	7:0	0	R/W	Only '0' allowed

Address (Hex)	Mnemonic	Bit	Reset Value	Type	Description
17	FIFO_STATUS				FIFO Status Register
	Reserved	7	0	R/W	Only '0' allowed
	TX_REUSE	6	0	R	Used for a PTX operation Pulse the r <sub>f</sub> ce high for at least 10 μs to Reuse last transmitted payload. TX payload reuse is active until w_TX_PAYLOAD or FLUSH TX is executed. TX_REUSE is set by the SPI command REUSE_TX_PL, and is reset by the SPI commands w_TX_PAYLOAD or FLUSH TX
	TX_FULL	5	0	R	TX FIFO full flag. 1: TX FIFO full. 0: Available locations in TX FIFO.
	TX_EMPTY	4	1	R	TX FIFO empty flag. 1: TX FIFO empty. 0: Data in TX FIFO.
	Reserved	3:2	00	R/W	Only '00' allowed
	RX_FULL	1	0	R	RX FIFO full flag. 1: RX FIFO full. 0: Available locations in RX FIFO.
	RX_EMPTY	0	1	R	RX FIFO empty flag. 1: RX FIFO empty. 0: Data in RX FIFO.
N/A	ACK_PLD	255:0	X	W	Written by separate SPI command ACK packet payload to data pipe number PPP given in SPI command. Used in RX mode only. Maximum three ACK packet payloads can be pending. Payloads with same PPP are handled first in first out.
N/A	TX_PLD	255:0	X	W	Written by separate SPI command TX data payload register 1 - 32 bytes. This register is implemented as a FIFO with three levels. Used in TX mode only.
N/A	RX_PLD	255:0	X	R	Read by separate SPI command. RX data payload register. 1 - 32 bytes. This register is implemented as a FIFO with three levels. All RX channels share the same FIFO.
1C	DYNPD				Enable dynamic payload length
	Reserved	7:2	0	R/W	Only '0' allowed
	DPL_P1	1	0	R/W	Enable dynamic payload length data pipe 1. (Requires EN_DPL and ENAA_P1)
	DPL_P0	0	0	R/W	Enable dynamic payload length data pipe 0. (Requires EN_DPL and ENAA_P0)
1D	FEATURE			R/W	Feature Register
	Reserved	7:3	0	R/W	Only '00000' allowed
	EN_DPL	2	0	R/W	Enables Dynamic Payload Length
	EN_ACK_PAY <sup>4</sup>	1	0	R/W	Enables Payload with ACK

Address (Hex)	Mnemonic	Bit	Reset Value	Type	Description
	EN_DYN_ACK	0	0	R/W	Enables the W_TX_PAYLOAD_NOACK command

1. Please take care when setting this parameter. If the ACK payload is more than 15 byte in 2 Mbps mode the ARD must be 500  $\mu$ s or more, if the ACK payload is more than 5 byte in 1 Mbps mode the ARD must be 500  $\mu$ s or more. In 250 kbps mode (even when the payload is not in ACK) the ARD must be 500  $\mu$ s or more.
2. This is the time the PTX is waiting for an ACK packet before a retransmit is made. The PTX is in RX mode for a minimum of 250  $\mu$ s, but it stays in RX mode to the end of the packet if that is longer than 250  $\mu$ s. Then it goes to standby-I mode for the rest of the specified ARD. After the ARD it goes to TX mode and then retransmits the packet.
3. The RX\_DR IRQ is asserted by a new packet arrival event. The procedure for handling this interrupt should be: 1) read payload through SPI, 2) clear RX\_DR IRQ, 3) read FIFO\_STATUS to check if there are more payloads available in RX FIFO, 4) if there are more data in RX FIFO, repeat from step 1).
4. If ACK packet payload is activated, ACK packets have dynamic payload lengths and the Dynamic Payload Length feature should be enabled for pipe 0 on the PTX and PRX. This is to ensure that they receive the ACK packets with payloads. If the ACK payload is more than 15 byte in 2 Mbps mode the ARD must be 500  $\mu$ s or more, and if the ACK payload is more than 5 byte in 1 Mbps mode the ARD must be 500  $\mu$ s or more. In 250 kbps mode (even when the payload is not in ACK) the ARD must be 500  $\mu$ s or more.

**Table 11** Register map of the RF transceiver

## 4 MCU

nRF31512 contains a fast 8-bit MCU, which executes the normal 8051 instruction set.

The architecture eliminates redundant bus states and implements parallel execution of fetch and execution phases. Most of the one-byte instructions are performed in a single cycle. The MCU uses one clock per cycle. This leads to a performance improvement rate of 8.0 (in terms of MIPS) with respect to legacy 8051 devices.

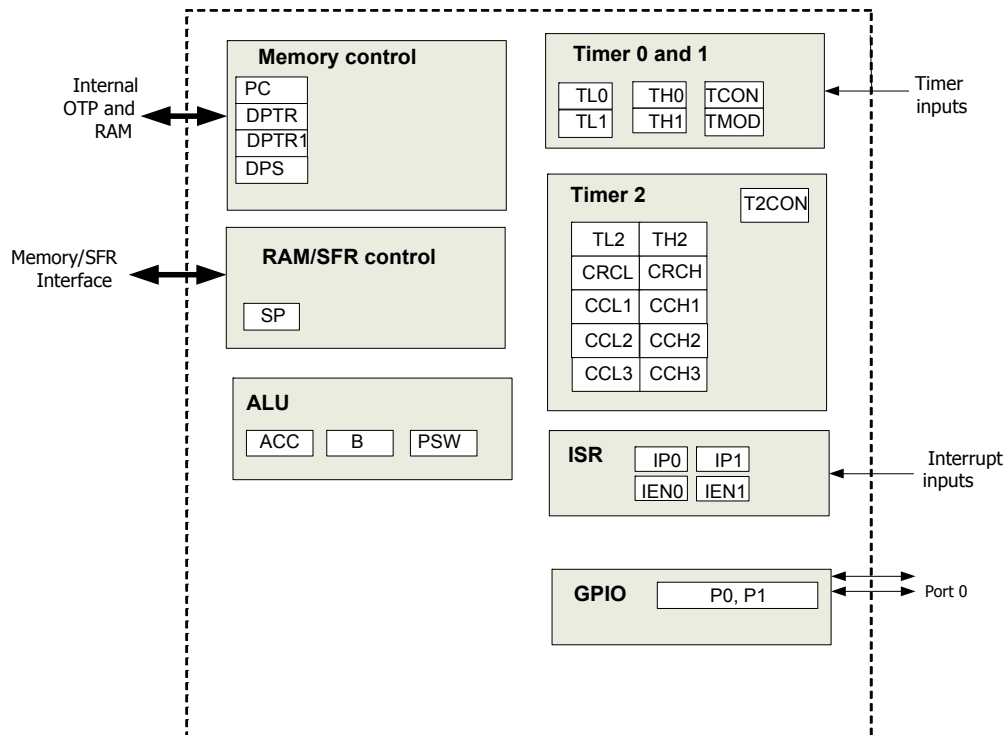
The original 8051 had a 12 clock architecture. A machine cycle needed 12 clocks and most instructions were either one or two machine cycles. Except for MUL and DIV instructions, the 8051 used either 12 or 24 clocks for each instruction. Each cycle in the 8051 also used two memory fetches. In many cases, the second fetch was a dummy, and extra clocks were wasted.

**Table 12** shows the speed advantage compared to a legacy 8051. A speed advantage of 12 implies that the instruction is executed twelve times faster. The average speed advantage is 8.0. However, the real speed improvement seen in any system depends on the instruction mix.

Speed advantage	Number of instructions	Number of opcodes
24	1	1
12	27	83
9.6	2	2
8	16	38
6	44	89
4.8	1	2
4	18	31
3	2	9
Average: 8.0	Sum: 111	Sum: 255

**Table 12** Speed advantage summary

## 4.1 Block diagram



**Figure 27** MCU block diagram

## 4.2 Features

- Control Unit
- 8-bit instruction decoder
- Reduced instruction cycle time (up to 12 times in respect to standard 80C51)
  - Arithmetic-Logic Unit
- 8-bit arithmetic and logical operations
- Boolean manipulations
- 8x8 bit multiplication and 8 / 8 bit division
  - Three 16-bit Timers/Counters
- 80C51-like Timer 0 & 1
- 80515-like Timer 2
  - Compare/Capture Unit, dedicated to Timer 2
- Software control capture
  - Interrupt Controller
- Four Priority Levels with:
- 13 interrupt sources for nRF31512
- 16-bit address bus
- Dual Data Pointer for fast data block transfer

## 4.3 Functional description

### 4.3.1 Arithmetic Logic Unit (ALU)

The Arithmetic Logic Unit (ALU) provides 8-bit division, 8-bit multiplication, and 8-bit addition with or without carry. The ALU also provides 8-bit subtraction with borrow and some bitwise logic operations, that is, logical AND, OR, Exclusive OR or NOT.

All operations are unsigned integer operations. Additionally, the ALU can increment or decrement 8-bit registers. For accumulator only, it can rotate left or right through carry or not, swap nibbles, clear or complement bits and perform a decimal adjustment. The ALU is handled by three registers, which are memory mapped as special function registers. Operands for operations may come from accumulator ACC, register B or from outside of the unit. The result may be stored in accumulator ACC or may be driven outside of the unit. The control register, that contains flags such as carry, overflow or parity, is the PSW (Program Status Word) register.

### 4.3.2 Instruction set summary

All instructions are binary code compatible and perform the same functions as they do within the legacy 8051 processor. The following tables give a summary of the instruction set with the required corresponding clock cycles.

Mnemonic	Description	Code	Bytes	Cycles
ADD A,Rn	Add register to accumulator	0x28-0x2F	1	1
ADD A,direct	Add directly addressed data to accumulator	0x25	2	2
ADD A,@Ri	Add indirectly addressed data to accumulator	0x26-0x27	1	2
ADD A,#data	Add immediate data to accumulator	0x24	2	2
ADDC A,Rn	Add register to accumulator with carry	0x38-0x3F	1	1
ADDC A, direct	Add directly addressed data to accumulator with carry	0x35	2	2
ADDC A,@Ri	Add indirectly addressed data to accumulator with carry	0x36-0x37	1	2
ADDC A,#data	Add immediate data to accumulator with carry	0x34	2	2
SUBB A,Rn	Subtract register from accumulator with borrow	0x98-0x9F	1	1
SUBB A, direct	Subtract directly addressed data from accumulator with borrow	0x95	2	2
SUBB A, @Ri	Subtract indirectly addressed data from accumulator with borrow	0x96-0x97	1	2
SUBB A, #data	Subtract immediate data from accumulator with borrow	0x94	2	2
INC A	Increment accumulator	0x04	1	1
INC Rn	Increment register	0x08-0x0F	1	2
INC direct	Increment directly addressed location	0x05	2	3
INC @Ri	Increment indirectly addressed location	0x06-0x07	1	3
INC DPTR	Increment data pointer	0xA3	1	1
DEC A	Decrement accumulator	0x14	1	1
DEC Rn	Decrement register	0x18-0x1F	1	2
DEC direct	Decrement directly addressed location	0x15	2	3
DEC @Ri	Decrement indirectly addressed location	0x16-0x17	1	3
MUL AB	Multiply A and B	0xA4	1	5
DIV	Divide A by B	0x84	1	5
DA A	Decimal adjust accumulator	0xD4	1	1

*Table 13 Arithmetic operations*

Mnemonic	Description	Code	Bytes	Cycles
ANL A, Rn	AND register to accumulator	0x58-0x5F	1	1
ANL A,direct	AND directly addressed data to accumulator	0x55	2	2
ANL A,@Ri	AND indirectly addressed data to accumulator	0x56-0x57	1	2
ANL A,#data	AND immediate data to accumulator	0x54	2	2
ANL direct,A	AND accumulator to directly addressed location	0x52	2	3
ANL direct,#data	AND immediate data to directly addressed location	0x53	3	4
ORL A,Rn	OR register to accumulator	0x48-0x4F	1	1
ORL A,direct	OR directly addressed data to accumulator	0x45	2	2
ORL A,@Ri	OR indirectly addressed data to accumulator	0x46-0x47	1	2
ORL A,#data	OR immediate data to accumulator	0x44	2	2
ORL direct,A	OR accumulator to directly addressed location	0x42	2	3
ORL direct,#data	OR immediate data to directly addressed location	0x43	3	4
XRL A,Rn	Exclusive OR register to accumulator	0x68-0x6F	1	1
XRL A, direct	Exclusive OR directly addressed data to accumulator	0x65	2	2
XRL A,@Ri	Exclusive OR indirectly addressed data to accumulator	0x66-0x67	2	2
XRL A,#data	Exclusive OR immediate data to accumulator	0x64	2	2
XRL direct,A	Exclusive OR accumulator to directly addressed location	0x62	2	3
XRL direct,#data	Exclusive OR immediate data to directly addressed location	0x63	3	4
CLR A	Clear accumulator	0xE4	1	1
CPL A	Complement accumulator	0xF4	1	1
RL A	Rotate accumulator left	0x23	1	1
RLC A	Rotate accumulator left through carry	0x33	1	1
RR A	Rotate accumulator right	0x03	1	1
RRC A	Rotate accumulator right through carry	0x13	1	1
SWAP A	Swap nibbles within the accumulator	0xC4	1	1

**Table 14** Logic operations



Mnemonic	Description	Code	Bytes	Cycles
MOV A,Rn	Move register to accumulator	0xE8-0xEF	1	1
MOV A,direct	Move directly addressed data to accumulator	0xE5	2	2
MOV A,@Ri	Move indirectly addressed data to accumulator	0xE6-0xE7	1	2
MOV A,#data	Move immediate data to accumulator	0x74	2	2
MOV Rn,A	Move accumulator to register	0xF8-0xFF	1	2
MOV Rn,direct	Move directly addressed data to register	0xA8-0xAF	2	4
MOV Rn,#data	Move immediate data to register	0x78-0x7F	2	2
MOV direct,A	Move accumulator to direct	0xF5	2	3
MOV direct,Rn	Move register to direct	0x88-0x8F	2	3
MOV direct1,direct2	Move directly addressed data to directly addressed location	0x85	3	4
MOV direct,@Ri	Move indirectly addressed data to directly addressed location	0x86-0x87	2	4
MOV direct,#data	Move immediate data to directly addressed location	0x75	3	3
MOV @Ri,A	Move accumulator to indirectly addressed location	0xF6-0xF7	1	3
MOV @Ri,direct	Move directly addressed data to indirectly addressed location	0xA6-0xA7	2	5
MOV @Ri,#data	Move immediate data to indirectly addressed location	0x76-0x77	2	3
MOV DPTR,#data16	Load data pointer with a 16-bit immediate	0x90	3	3
MOVC A,@A+DPTR	Load accumulator with a code byte relative to DPTR	0x93	1	3
MOVC A,@A+PC	Load accumulator with a code byte relative to PC	0x83	1	3
MOVX A,@Ri	Move <sup>1</sup> external RAM (8-bit addr) to accumulator	0xE2-0xE3	1	4
MOVX A,@DPTR	Move <sup>1</sup> external RAM (16-bit addr) to accumulator	0xE0	1	4
MOVX @Ri,A	Move <sup>1</sup> accumulator to external RAM (8-bit addr)	0xF2-0xF3	1	5
MOVX @DPTR,A	Move <sup>1</sup> accumulator to external RAM (16-bit addr)	0xF0	1	5
PUSH direct	Push directly addressed data onto stack	0xC0	2	4
POP direct	Pop directly addressed location from stack	0xD0	2	3
XCH A,Rn	Exchange register with accumulator	0xC8-0xCF	1	2
XCH A,direct	Exchange directly addressed location with accumulator	0xC5	2	3
XCH A,@Ri	Exchange indirect RAM with accumulator	0xC6-0xC7	1	3
XCHD A,@Ri	Exchange low-order nibbles of indirect and accumulator	0xD6-0xD7	1	3

1. The MOVX instructions perform one of two actions depending on the state of **pmw** bit (pcon.4).

**Table 15** Data transfer operations

Mnemonic	Description	Code	Bytes	Cycles
ACALL addr11	Absolute subroutine call	xxx10001b	2	6
LCALL addr16	Long subroutine call	0x12	3	6
RET	Return from subroutine	0x22	1	4
RETI	Return from interrupt	0x32	1	4
AJMP addr11	Absolute jump	xxx00001b	2	3
LJMP addr16	Long jump	0x02	3	4
SJMP rel	Short jump (relative address)	0x80	2	3
JMP @A+DPTR	Jump indirect relative to the DPTR	0x73	1	2
JZ rel	Jump if accumulator is zero	0x60	2	3
JNZ rel	Jump if accumulator is not zero	0x70	2	3
JC rel	Jump if carry flag is set	0x40	2	3
JNC rel	Jump if carry flag is not set	0x50	2	3
JB bit, rel	Jump if directly addressed bit is set	0x20	3	4
JNB bit, rel	Jump if directly addressed bit is not set	0x30	3	4
JBC bit, rel	Jump if directly addressed bit is set and clear bit	0x10	3	4
CJNE A, direct, rel	Compare directly addressed data to accumulator and jump if not equal	0xB5	3	4
CJNE A, #data, rel	Compare immediate data to accumulator and jump if not equal	0xB4	3	4
CJNE Rn, #data, rel	Compare immediate data to register and jump if not equal	0xB8–0xBF	3	4
CJNE @Ri, #data, rel	Compare immediate data to indirect addressed value and jump if not equal	0xB6–B7	3	4
DJNZ Rn, rel	Decrement register and jump if not zero	0xD8–DF	2	3
DJNZ direct, rel	Decrement directly addressed location and jump if not zero	0xD5	3	4
NOP	No operation	0x00	1	1

Table 16 Program branches

Mnemonic	Description	Code	Bytes	Cycles
CLR C	Clear carry flag	0xC3	1	1
CLR bit	Clear directly addressed bit	0xC2	2	3
SETB C	Set carry flag	0xD3	1	1
SETB bit	Set directly addressed bit	0xD2	2	3
CPL C	Complement carry flag	0xB3	1	1
CPL bit	Complement directly addressed bit	0xB2	2	3
ANL C, bit	AND directly addressed bit to carry flag	0x82	2	2
ANL C, /bit	AND complement of directly addressed bit to carry	0xB0	2	2
ORL C, bit	OR directly addressed bit to carry flag	0x72	2	2
ORL C, /bit	OR complement of directly addressed bit to carry	0xA0	2	2

Mnemonic		Description	Code	Bytes	Cycles
MOV C,bit		Move directly addressed bit to carry flag	0xA2	2	2
MOV bit,C		Move carry flag to directly addressed bit	0x92	2	3

*Table 17 Boolean manipulation*

### 4.3.3 Opcode map

Opcode	Mnemonic	Opcode	Mnemonic	Opcode	Mnemonic
00H	NOP	56H	ANL A,@R0	ACH	MOV R4,direct
01H	AJMP addr11	57H	ANL A,@R1	ADH	MOV R5,direct
02H	JUMP addr16	58H	ANL A,R0	AE	MOV R6,direct
03H	RRA	59H	ANL A,R1	AFH	MOV R7,direct
04H	INCA	5AH	ANL A,R2	B0H	ANL C,/bit
05H	INC direct	5BH	ANL A,R3	B1H	ACALL addr11
06H	INC @R0	5CH	ANL A,R4	B2H	CPL bit
07H	INC @R1	5DH	ANL A,R5	B3H	CPLC
08H	INC R0	5EH	ANL A,R6	B4H	CJNE A,#data,rel
09H	INC R1	5FH	ANL A,R7	B5H	CJNE A, direct, rel
0AH	INC R2	60H	JZ rel	B6H	CJNE @R0,#data,rel
0BH	INC R3	61H	AJMP addr11	B7H	CJNE @R1, #data,rel
0CH	INC R4	62H	XRL direct, A	B8H	CJNE R0, #data,rel
0DH	INC R5	63H	XRL direct, #data	B9H	CJNE R1,#data,rel
0EH	INC R6	64H	XRL A, #data	BAH	CJNE R2,#data,rel
0FH	INC R7	65H	XRL A,direct	BBH	CJNE R3,#data,rel
10H	JBC bit, rel	66H	XRLA,@R0	BCH	CJNE R4,#data,rel
11H	ACALL addr11	67H	XRL A,@R1	BDH	CJNE R5,#data,rel
12H	LCALL add r16	68H	XRL A,R0	BEH	CJNE R6,#data,rel
13H	RRC A	69H	XRL A,R1	BFH	CJNE R7,#data,rel
14H	DEC A	6AH	XRL A,R2	C0H	PUSH direct
15H	DEC direct	6BH	XRL A,R3	C1H	AJMP addr11
16H	DEC @R0	6CH	XRL A,R4	C2H	CLR bit
17H	DEC @R1	6DH	XRL A,R5	C3H	CLR C
18H	DEC R0	6EH	XRL A,R6	C4H	SWAP A
19H	DEC R1	6FH	XRL A,R7	C5H	XCH A, direct
1AH	DEC R2	70H	JNZ rel	C6H	XCH A,@R0
1BH	DECR3	71H	ACALL addr11	C7H	XCH A,@R1
1CH	DECR4	72H	ORL C, bit	C8H	XCH A,R0
1DH	DECR5	73H	JMP @A+DPTR	C9H	XCH A,R1
1EH	DECR6	74H	MOV A, #data	CAH	XCH A,R2
1FH	DECR7	75H	MOV direct, #data	CBH	XCHA,R3
20H	JB bit, rel	76H	MOV @R0,#data	CCH	XCH A,R4
21H	AJMP addr11	77H	MOV @R1, #data	CDH	XCH A,R5
22H	RET	78H	MOV R0, #data	CEH	XCH A,R6
23H	RL A	79H	MOV R1, #data	CFH	XCHA,R7
24H	ADD A, #data	7AH	MOV R2, #data	D0H	POP direct
25H	ADD A, direct	7BH	MOV R3, #data	D1H	ACALL addr11
26H	ADD A,@R0	7CH	MOV R4, #data	D2H	SETB bit

Opcode	Mnemonic	Opcode	Mnemonic	Opcode	Mnemonic
27H	ADD A,@R1	7DH	MOV R5, #data	D3H	SETB C
28H	ADD A,R0	7EH	MOV R6, #data	D4H	DAA
29H	ADD A,R1	7FH	MOV R7, #data	D5H	DJNZ direct, rel
2AH	ADD A,R2	80H	SJMP rel	D6H	XCHDA,@R0
2BH	ADD A,R3	81H	AJMP addr11	D7H	XCHD A,@R1
2CH	ADD A,R4	82H	ANL C, bit	D8H	DJNZ R0,rel
2DH	ADD A,R5	83H	MOVC A,@A+PC	D9H	DJNZ R1,rel
2EH	ADD A,R6	84H	DIV AB	DAH	DJNZ R2,rel
2FH	ADD A,R7	85H	MOV direct, direct	DBH	DJNZ R3,rel
30H	JNB bit, rel	86H	MOV direct,@R0	DCH	DJNZ R4,rel
31H	ACALL addr11	87H	MOV direct,@R1	DDH	DJNZ R5,rel
32H	RETI	88H	MOV direct,R0	DE	DJNZ R6,rel
33H	RLC A	89H	MOV direct,R1	DFH	DJNZ R7,rel
34H	ADDC A,#data	8AH	MOV direct,R2	E0H	MOVX A,@DPTR
35H	ADDC A, direct	8BH	MOV direct,R3	E1H	AJMP addr11
36H	ADDC A,@R0	8CH	MOV direct,R4	E2H	MOVX A,@R0
37H	ADDC A,@R1	8DH	MOV direct, R5	E3H	MOVX A,@R1
38H	ADDC A,R0	8EH	MOV direct,R6	E4H	CLR A
39H	ADDC A,R1	8FH	MOV direct,R7	E5H	MOVA, direct
3AH	ADDC A,R2	90H	MOV DPTR, #data16	E6H	MOVA,@R0
3BH	ADDC A,R3	91H	ACALL addr11	E7H	MOV A,@R1
3CH	ADDC A,R4	92H	MOV bit, C	E8H	MOV A,R0
3DH	ADDC A,R5	93H	MOVCA,@A+DPTR	E9H	MOV A,R1
3EH	ADDC A,R6	94H	SUBB A, #data	EAH	MOV A,R2
3FH	ADDC A,R7	95H	SUBB A, direct	EBH	MOV A,R3
40H	JC rel	96H	SUBB A,@R0	ECH	MOV A,R4
41H	AJMP addr11	97H	SUBB A,@R1	EDH	MOV A,R5
42H	ORL direct, A	98H	SUBB A, R0	EEH	MOV A,R6
43H	ORL direct, #data	99H	SUBB A,R1	EFH	MOV A,R7
44H	ORL A, #data	9AH	SUBB A,R2	F0H	MOVX @DPTR,A
45H	ORL A, direct	9BH	SUBB A,R3	F1H	ACALL addr11
46H	ORL A,@R0	9CH	SUBB A,R4	F2H	MOVX @R0,A
47H	ORL A,@R1	9DH	SUBB A,R5	F3H	MOVX @R1,A
48H	ORL A,R0	9EH	SUBB A,R6	F4H	CPL A
49H	ORL A,R1	9FH	SUBB A,R7	F5H	MOV direct, A
4AH	ORL A,R2	A0H	ORL C,/bit	F6H	MOV @R0,A
4BH	ORLA,R3	A1H	AJMP addr11	F7H	MOV @R1,A
4CH	ORL A,R4	A2H	MOV C, bit	F8H	MOV R0,A
4DH	ORL A,R5	A3H	INC DPTR	F9H	MOV R1,A
4EH	ORL A,R6	A4H	MUL AB	FAH	MOV R2,A
4FH	ORLA,R7	A5H	-	FBH	MOV R3,A

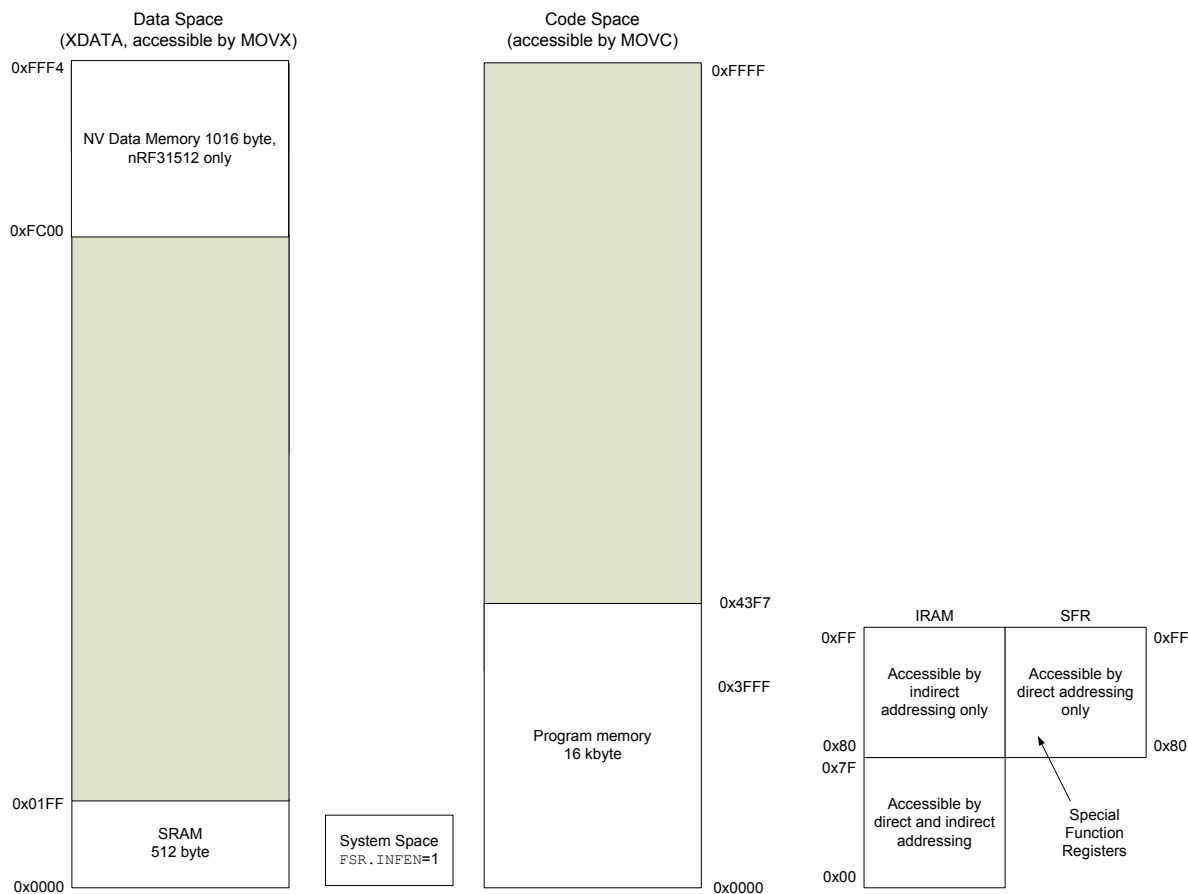
Opcode	Mnemonic	Opcode	Mnemonic	Opcode	Mnemonic
50H	JNC rel	A6H	MOV @R0,direct	FCH	MOV R4,A
51H	ACALL addr11	A7H	MOV @R1,direct	FDH	MOV R5,A
52H	ANL direct, A	A8H	MOV R0,direct	FEH	MOV R6,A
53H	ANL direct, #data	A9H	MOV R1,direct	FFH	MOV R7,A
54H	ANL A, #data	AAH	MOV R2,direct		
55H	ANL A, direct	ABH	MOV R3,direct		

*Table 18 Opcode map*

## 5 Memory and I/O organization

The MCU has 64 kB of separate address space for code and data, an area of 256 bytes for internal data (IRAM) and an area of 128 bytes for Special Function Registers (SFR), and some System Data in system data space.

The nRF31512 has 17397 bytes of OTP memory usable for code and data. It also has 512 bytes of data memory (SRAM). See default memory map in **Figure 28**. Read- or write access to the grey areas in this figure may behave unpredictably.



**Figure 28** Memory map

The lower 128 bytes of the IRAM contains work registers (0x00–0x1F) and bit addressable memory (0x20–0x2F). The upper half can only be accessed by indirect addressing.

The lowest 32 bytes of the IRAM form four banks, each consisting of eight registers (R0–R7). Two bits of the program memory status word (PSW) select which bank is used. The next 16 bytes of memory form a block of bit-addressable memory, accessible through bit addresses 0x00–0x7F.

## 5.1 System space

The INFEN bit in the FSR register can be set to force data fetches from the code space to a system data space that contains IC specific information generated during manufacturing.

When the FSR.INFEN bit is set, the program cannot access any regular variables in the code space, so the reading of system space variables should use the following procedure:

1. Disable all interrupts if required (save EA and set EA=0)
2. Set FSR.INFEN=1
3. Read data from system space (do not use C library functions when FSR.INFEN=1.
4. Set FSR.INFEN=0
5. Restore interrupts if required (restore EA from save value)

Address	Bit	R/W	Function	Reset value 0x00
0xF8	7:4	R	See section 6.3 on page 72 for information on usage.	
	3	R/W	INFEN, enable access to System Space	
	2:0	R	Ignore	

*Table 19 INFEN bit in the FSR register*

The following table contains a description of the system data space variables.

Variable	Size	Address	Comment
CHIPID	4 bytes	0x000C	ID number for each individual device. The ID is generated by a pseudo random process. No ID will violate the rules specified in section 3.4.3.2 on page 22

*Table 20 System space variables*

## 5.2 PDATA memory addressing

The MCU supports PDATA (Paged Data memory) addressing into data space. One page (256 bytes) can be accessed by an indirect addressing scheme through registers R0 and R1 (@R0, @R1).

The MPAGE register controls the start address of the PDATA page:

Addr	Bit	R/W	Function	Reset value: 0x00
0xC9	7:0	R/W	Start address of the PDATA page	

*Table 21 MPAGE register*



MPAGE sets the upper half of the 16 bit address space. For example, setting MPAGE to 0x80 starts PDATA from address 0x8000.

## 5.3 MCU Special Function Registers

### 5.3.1 Accumulator - ACC

Accumulator is used by most of the MCU instructions to hold the operand and to store the result of an operation. The mnemonics for accumulator specific instructions refer to accumulator as A, not ACC.

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0xE0	acc.7	acc.6	acc.5	acc.4	acc.3	acc.2	acc.1	acc.0

Table 22 ACC register

### 5.3.2 B Register – B

The B register is used during multiplying and division instructions. It can also be used as a scratch-pad register to hold temporary data.

Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0xF0	b.7	b.6	b.5	b.4	b.3	b.2	b.1	b.0

Table 23 B register

### 5.3.3 Program Status Word Register - PSW

The PSW register contains status bits that reflect the current state of the MCU.

**Note:** The Parity bit can only be modified by hardware upon the state of ACC register.

Address	Bit	Name	Description
0xD0	7	cy	Carry flag: Carry bit in arithmetic operations and accumulator for Boolean operations.
	6	ac	Auxiliary Carry flag: Set if there is a carry-out from 3rd bit of Accumulator in BCD operations
	5	f0	General purpose flag 0
	4:3	rs	Register bank select, bank 0..3 (0x00–0x07, 0x08–0x0f, 0x10–0x17, 0x18–0x1f)
	2	ov	Overflow flag: Set if overflow in Accumulator during arithmetic operations
	1	f1	General purpose flag 1
	0	p	Parity flag: Set if odd number of '1' in ACC.

Table 24 PSW register

### 5.3.4 Stack Pointer – SP

This register points to the top of stack in internal data memory space. It is used to store the return address of a program before executing interrupt routine or subprograms. The SP is incremented before executing PUSH or CALL instruction and it is decremented after executing POP or RET(I) instruction (it always points to the top of stack).

Address	Register name
0x81	SP

*Table 25 SP register*

### 5.3.5 Data Pointer – DPH, DPL

Address	Register name
0x82	DPL
0x83	DPH

*Table 26 Data Pointer register (DPH:DPL)*

The Data Pointer Registers can be accessed through DPL and DPH. The actual data pointer is selected by DPS register.

These registers are intended to hold 16-bit address in the indirect addressing mode used by MOVX (move external memory), MOVC (move program memory) or JMP (computed branch) instructions. They may be manipulated as 16-bit register or as two separate 8-bit registers. DPH holds higher byte and DPL holds lower byte of indirect address.

It is generally used to access external code or data space (for example, MOVC A, @A+DPTR or MOV A, @DPTR respectively).

### 5.3.6 Data Pointer 1 – DPH1, DPL1

Address	Register name
0x84	DPL1
0x85	DPH1

*Table 27 Data Pointer 1 register (DPH1:DPL1)*

The Data Pointer Register 1 can be accessed through DPL1 and DPH1. The actual data pointer is selected by DPS register.

These registers are intended to hold 16-bit address in the indirect addressing mode used by MOVX (move external memory), MOVC (move program memory) or JMP (computed branch) instructions. They may be manipulated as 16-bit register or as two separate 8-bit registers. DPH1 holds higher byte and DPL1 holds lower byte of indirect address.

It is generally used to access external code or data space (for example, MOVC A<sub>i</sub>@A+DPTR or MOV A<sub>i</sub>@DPTR respectively).

The Data Pointer 1 is an extension to the standard 8051 architecture to speed up block data transfers.

### 5.3.7 Data Pointer Select Register – DPS

The MCU contains two Data Pointer registers. Both of them can be used as 16-bits address source for indirect addressing. The DPS register serves for selecting active data pointer register.

Address	Bit	Name	Description
0x92	7:1	-	Not used
	0	dps	Data Pointer Select. 0: select DPH:DPL, 1: select DPH1:DPL1

*Table 28 DPS register*

### 5.3.8 PCON register

The PCON register is used to control the Program Memory Write Mode and Serial Port 0 baud rate doubler.

Address	Bit	Name	Description
0x87	7	smod	Serial port 0 baud rate select
	6	gf3	General purpose flag 3
	5	gf2	General purpose flag 2
	4	pmw	Program memory write mode: 1: MOVX instructions will access memory code space 0: MOVX instructions will access memory data space
	3	gf1	General purpose flag 1
	2	gfo	General purpose flag 0
	1	-	Not used. This bit must always be cleared. Always read as 0.
	0	-	Not used. This bit must always be cleared. Always read as 0.

*Table 29 PCON register*

### 5.3.9 Special Function Register Map

The map of Special Function Registers is shown in *Table 30*. Undefined locations must not be read or written.

Address	X000	X001	X010	X011	X100	X101	X110	X111
0xF8–0xFF	<i>FSR</i>	Reserved	Reserved	Reserved	<i>SPIMCON0</i>	<i>SPIMCON1</i>	<i>SPIMSTAT</i>	<i>SPIMDAT</i>
0xF0–0xF7	<i>B</i>	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0xE8–0xEF	<i>RFCON</i>	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0xE0–0xE7	<i>ACC</i>	Reserved	Reserved	<i>RTC2PCO</i>	<i>SPIRCON0</i>	<i>SPIRCON1</i>	<i>SPIRSTAT</i>	<i>SPIRDAT</i>
0xD8–0xDF	Reserved	Reserved	Reserved	Reserved	<i>POFCON</i>	<i>CCPDATIA</i>	<i>CCPDATIB</i>	<i>CCPDATO</i>
0xD0–0xD7	<i>PSW</i>	<i>ADCCON3</i>	Reserved	<i>ADCCON1</i>	<i>ADCDATH</i>	<i>ADCDATL</i>	Reserved	Reserved

Address	X000	X001	X010	X011	X100	X101	X110	X111
0xC8–0xCF	<i>T2CON</i>	<i>MPAGE</i>	<i>CRCL</i>	<i>CRCH</i>	<i>TL2</i>	<i>TH2</i>	<i>WUOPC1</i>	<i>WUOPC0</i>
0xC0–0xC7	<i>IRCON</i>	<i>CCEN</i>	<i>CCL1</i>	<i>CCH1</i>	<i>CCL2</i>	<i>CCH2</i>	<i>CCL3</i>	<i>CCH3</i>
0xB8–0xBF	<i>IEN1</i>	<i>IP1</i>	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0xB0–0xB7	Reserved	<i>RSTREAS</i>	Reserved	<i>RTC2CON</i>	<i>RTC2CMP0</i>	<i>RTC2CMP1</i>	<i>RTC2CPT00</i>	Reserved
0xA8–0xAF	<i>IEN0</i>	<i>IP0</i>	Reserved	<i>RTC2CPT01</i>	<i>RTC2CPT10</i>	<i>CLKLCTRL</i>	<i>OPMCON</i>	<i>WDSV</i>
0xA0–0xA7	Reserved	Reserved	Reserved	<i>CLKCTRL</i>	<i>PWRDWN</i>	<i>WUCON</i>	<i>INTEXP</i>	<i>MEMCON</i>
0x98–0x9F	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	<i>P0CON</i>	<i>P1CON</i>
0x90–0x97	<i>P1</i>	Reserved	<i>DPS</i>	<i>P0DIR</i>	<i>P1DIR</i>	Reserved	Reserved	Reserved
0x88–0x8F	<i>TCON</i>	<i>TMOD</i>	<i>TL0</i>	<i>TL1</i>	<i>TH0</i>	<i>TH1</i>	Reserved	Reserved
0x80–0x87	<i>P0</i>	<i>SP</i>	<i>DPL</i>	<i>DPH</i>	<i>DPL1</i>	<i>DPH1</i>	Reserved	<i>PCON</i>

**Table 30** *Special Function Registers locations*

The registers in the X000 column in B register are both byte and bit addressable. The other registers are only byte addressable.

### 5.3.10 Special Function Registers reset values

Register name	Address	Reset value	Description
ACC	0xE0	0x00	Accumulator
ADCCON1	0xD3	0x00	ADC Configuration Register 1
ADCCON3	0xD1	0x00	ADC Configuration Register 3
ADCDATH	0xD4	0x00	ADC Data high byte
ADCDATL	0xD5	0x00	ADC Data low byte
B	0xF0	0x00	B Register
CCEN	0xC1	0x00	Compare/Capture Enable Register
CCH1	0xC3	0x00	Compare/Capture Register 1, high byte
CCH2	0xC5	0x00	Compare/Capture Register 2, high byte
CCH3	0xC7	0x00	Compare/Capture Register 3, high byte
CCL1	0xC2	0x00	Compare/Capture Register 1, low byte
CCL2	0xC4	0x00	Compare/Capture Register 2, low byte
CCL3	0xC6	0x00	Compare/Capture Register 3, low byte
CCPDATIA	0xDD	0x00	Encryption/Decryption accelerator Data In Register A
CCPDATIB	0xDE	0x00	Encryption/Decryption accelerator Data In Register B
CCPDATO	0xDF	0x00	Encryption/Decryption accelerator Data Out Register
CLKLFCTRL	0xAD	0x07	32 kHz (CLKLF) control
CLKCTRL	0xA3	0x00	Clock control
CRCH	0xCB	0x00	Compare/Reload/Capture Register, high byte
CRCL	0xCA	0x00	Compare/Reload/Capture Register, low byte
DPH	0x83	0x00	Data Pointer High 0
DPL	0x82	0x00	Data Pointer Low 0
DPH1	0x85	0x00	Data Pointer High 1
DPL1	0x84	0x00	Data Pointer Low 1
DPS	0x92	0x00	Data Pointer Select Register
FSR	0xF8		OTP Status Register
IEN0	0xA8	0x00	Interrupt Enable Register 0
IEN1	0xB8	0x00	Interrupt Priority Register / Enable Register 1
INTEXP	0xA6	0x01	Interrupt Expander Register
IPO	0xA9	0x00	Interrupt Priority Register 0
IP1	0xB9	0x00	Interrupt Priority Register 1
IRCON	0xC0	0x00	Interrupt Request Control Register
MEMCON	0xA7	0x00	Memory Configuration Register
MPAGE	0xC9	0x00	Start address of the PDATA page
OPMCON	0xAE	0x00	Operational Mode Control
P0	0x80	0xFF	Port 0 value
P0CON	0x9E	1	Port 0 Configuration Register
P0DIR	0x93	0xFF	Port 0 pin direction control
P1	0x90	0xFF	Port 1 value

Register name	Address	Reset value	Description
P1CON	0x9F	1	Port 1 Configuration Register
P1DIR	0x94	0xFF	Port 1 pin direction control
PCON	0x87	0x00	PCON Register
POFCON	0xDC	0x00	Power-fail Comparator Configuration Register
PSW	0xD0	0x00	Program Status Word
PWRDWN	0xA4	0x00	Power-down control
RFCON	0xE8	0x02	RF transceiver Control Register
RSTREAS	0xB1	0x00	Reset Reason Register
RTC2CMP0	0xB4	0xFF	RTC2 Compare Value Register 0
RTC2CMP1	0xB5	0xFF	RTC2 Compare Value Register 1
RTC2CON	0xB3	0x00	RTC2 Configuration Register
RTC2CPT00	0xB6	0x00	RTC2 Capture Value Register 00
RTC2CPT01	0xAB	0x00	RTC2 Capture Value Register 01
RTC2CPT10	0xAC	0x00	RTC2 Capture Value Register 10
RTC2PCO	0xE3	0x63	RTC2 Pre-start Time
SP	0x81	0x07	Stack Pointer
SPIMCON0	0xFC	0x02	SPI Master Configuration Register 0
SPIMDAT	0xFF	0x00	SPI Master Data Register
SPIMSTAT	0xFE	0x03	SPI Master Status Register
SPIRCON0	0xE4	0x01	RF transceiver SPI Master Configuration Register 0
SPIRCON1	0xE5	0x0F	RF transceiver SPI Master Configuration Register 1
SPIRDAT	0xE7	0x00	RF transceiver SPI Master Data Register
SPIRSTAT	0xE6	0x03	RF transceiver SPI Master Status Register
T2CON	0xC8	0x00	Timer 2 Control Register
TCON	0x88	0x00	Timer/Counter Control Register
TH0	0x8C	0x00	Timer 0, high byte
TH1	0x8D	0x00	Timer 1, high byte
TH2	0xCD	0x00	Timer 2, high byte
TL0	0x8A	0x00	Timer 0, low byte
TL1	0x8B	0x00	Timer 1, low byte
TL2	0xCC	0x00	Timer 2, low byte
TMOD	0x89	0x00	Timer Mode Register
WDSW	0xAF	0x00	Watchdog Start Value Register
WUCON	0xA5	0x00	Wakeup configuration register
WUOPC0	0xCF	0x00	Wakeup On Pin Configuration Register 0
WUOPC1	0xCE	0x00	Wakeup On Pin Configuration Register 1

1. This register is used to access port configuration of P0/P1. After reset, the input configuration for all ports will be 00 (digital input, no pull up/down) and the output configuration for all ports will be 000 (digital output normal drive strength).

**Table 31** Special Function Registers reset values

## 6 OTP memory

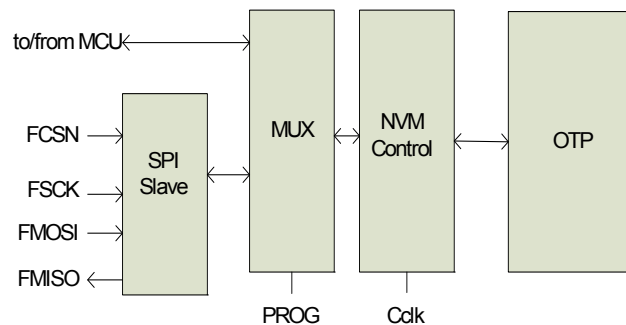
The nRF31512 has 17397 bytes of One-Time Programmable (OTP) memory intended as program and data memory. The primary use for this memory is for read-only program and data storage, but the MCU may also perform write operations.

The OTP memory can be programmed through an external SPI slave interface. The OTP memory can be configured to inhibit readback or modification of the memory content, and is organized in the same way as the nRF24LE1 device.

### 6.1 Features

- 17397 bytes OTP memory, code and data
- OTP programmable through SPI
- OTP read and write accessible from the MCU
- Configurable SPI readback protection

### 6.2 Block diagram



**Figure 29** OTP block diagram

## 6.3 Functional description

### 6.3.1 OTP memory configuration SFR

The FSR register is used to control the access of OTP memory and system data, and to read status of OTP.

Address (hex)	Mnemonic	Bit	Reset value	SPI access	SFR access	Description
0xF8	FSR					OTP Status Register
	-	7:6	0	R	R	Reserved
	WEN	5	0	R/W	R/W	OTP write enable latch. Enables OTP write operations from SPI.  WEN will be cleared after each SPI write operation, but not after a MCU operations.
	RDYN	4	1	R	R	OTP ready flag, active low.
	INFEN	3	0	R/W	R/W	System Data Enable. Enables access to system data, see also section <b>5.1 on page 64</b> .
	RDIS	2	0	R	R	OTP readback protection enabled. Can only be set once by use of SPI command RDISMB, and cannot be cleared again.
	-	1	0	R	R	Reserved
	-	0	0	R	R	Reserved

**Table 32** Registers for MCU and SPI for OTP configuration control

### 6.3.2 Memory compatibility with flash version of nRF24LE1

Regarding program size and memory configuration, nRF31512 is compatible with nRF24LE1 if program and data size are less or equal to 17397 bytes. See the *nRF24LE1 Product Specification* for details.

### 6.3.3 Brown-out

There is an on-chip power-fail brown-out detector, see **chapter 12 on page 109**, which ensures that any OTP memory program access will be ignored when the Power Fail (POF) signal, see **Figure 46 on page 109**, is active. Both the microcontroller and the OTP memory write operation still function according to specification, and any write operation that was started will be completed. The Power-fail comparator is disabled after startup and can be enabled by setting bit 7 in POFCON.

If the supply voltage drops below ~1.7V, that is when the Brown-Out Reset (BOR) signal (see **Figure 49 on page 111**) is active, the chip will be reset. If the power supply rises again before reaching the reset threshold, there will be no reset. In order to have an indication that shows this has happened, one will need to enable the Power Failure interrupt (POFIRQ, see **Table 47 on page 93**).



To ensure proper programming of the OTP in the cases where power supply may be unreliable, the user should take the following precautions:

- Make sure the data read back from the OTP is identical to what is written to OTP. The mechanism above will guarantee that the data is safely stored to OTP if the value does compare. If the compare fails, the write has been ignored due to a power supply event.
- Make sure that the time from “Power fail” to “Reset” is longer than one write operation (800 µs) by a sufficient reservoir on the supply.

### 6.3.4 Temperature restriction for OTP programming

The OTP program operations may be unreliable at temperatures below 0 °C.

### 6.3.5 OTP programming from the MCU

This section describes how you can write the OTP memory using the MCU.

#### 6.3.5.1 MCU operations in the main block

When an OTP write is initiated, the MCU is halted for 12800 clock cycles (800 µs) for each byte written.

The clock frequency of the microcontroller must be 16 MHz during OTP write operations.

To allow write OTP operations the MCU must run the following sequence:

1. Set WEN (bit 5) in the FSR register high to enable OTP write access. The OTP is now open for write from the MCU until WEN in FSR is set low again.
2. If write to code space is intended, set PMW (bit 4) in the PCON register high to enable program memory write mode.
3. Programming the OTP is done through normal memory write operations from the MCU. Bytes are written individually (there is no auto increment) to the OTP using the specific memory address.

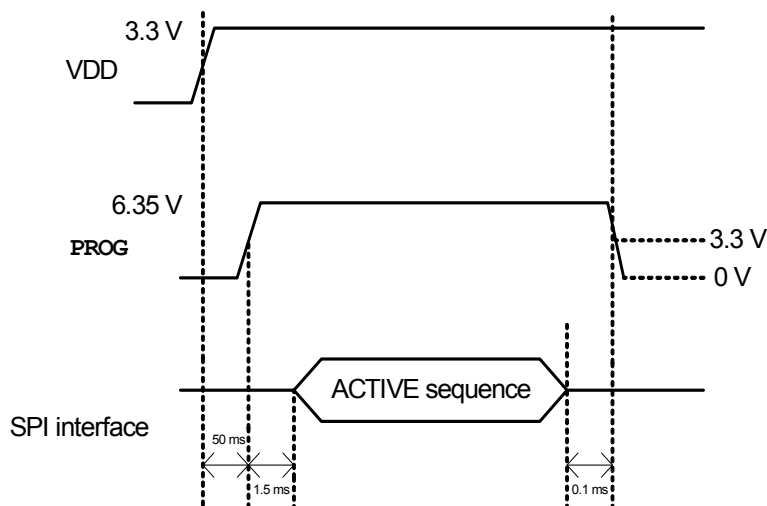
When the programming code executes from the OTP, write operation is self-timed and the CPU stops until the operation is finished. If the programming code executes from the XDATA RAM the code must wait until the operation has finished. This can be done either by polling the RDYN bit in the FSR register to go low or by a wait loop. Do not set WEN low before the write operation is finished. Memory address is identical to the OTP address, see *chapter 5 on page 63* for memory mapping.

### 6.3.6 OTP programming through SPI

The on-chip OTP is designed to interface a standard SPI device for programming. The interface uses an 8-bit command register and a set of commands to program and configure the OTP memory.

### 6.3.6.1 PROG pin requirements

During SPI programming of the OTP, the **PROG** pin is used as a high voltage supply.



**Figure 30** Waveform for OTP programming via SPI

The **PROG** pin is raised to VPP (6.35 V) minimum of 1.5 ms before first SPI command is issued. The **PROG** pins should be held at VPP for 0.1 ms after the last SPI command is completed. For SPI read the **PROG** pin supply may be reduced to 3.3V.

**Note:** An external pull-down resistor on the **PROG** pin is required. If external programming is not needed **PROG** must be connected to ground.

### 6.3.6.2 SPI slave interface

To program the memory an SPI slave interface is used. SPI slave connection to the OTP memory is activated by setting pin **PROG** = 1 while the reset pin is kept inactive. After the **PROG** pin is set to 1, apply a pulse on the **RESET** pin (Pull **RESET** pin low for a minimum of 0.2  $\mu$ s and return to high.) The **GPIO** pins are automatically configured as a SPI slave as shown in **Table 33** Further information on SPI slave timing can be found in **chapter 16 on page 126**.

	Pin
FCSN	P0 . 7
FMISO	P0 . 6
FMOSI	P0 . 5
FSCK	P0 . 4

**Table 33** OTP SPI slave physical interface

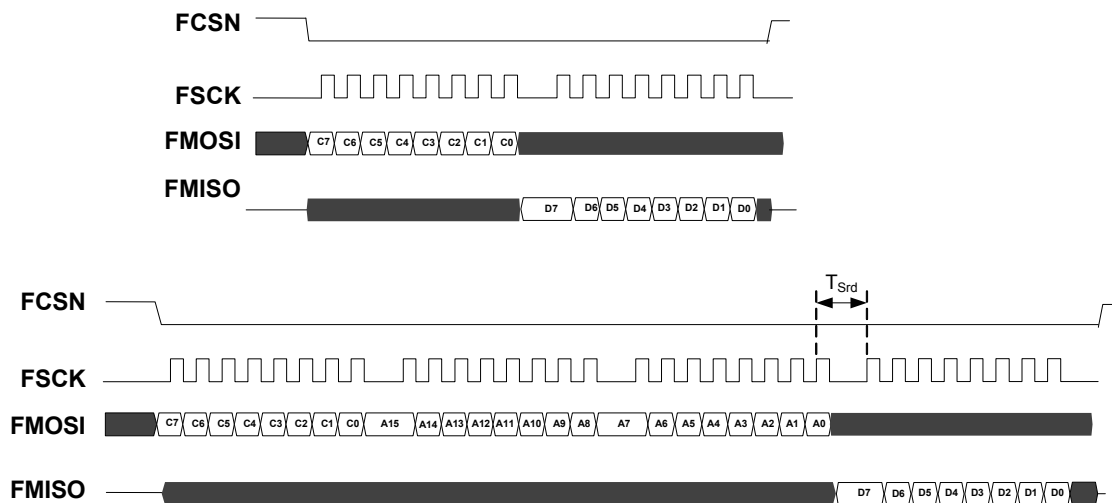
**Note:** After activation of the **PROG** pin you must wait at least 1.5 ms before you input the first OTP command.

The program interface uses an 8 bit command register and a set of commands to program and configure the OTP memory.

Command	Command format	Address	# Data bytes	Command operation
WREN	0x06	NA	0	Set OTP write enable latch. Bit WEN in register FSR
WRDIS	0x04	NA	0	Reset OTP write enable latch. Bit WEN in register FSR
RDSR	0x05	NA	1	Read OTP Status Register (FSR)
WRSR	0x01	NA	1	Write OTP Status Register (FSR)
READ	0x03	2 bytes, First OTP address to to be read	1–17408	Read data from OTP
PROGRAM	0x02	2 bytes, first OTP address to be written	1–1024	Write data to OTP  <b>Note:</b> WEN must be set.
RDISMB	0x85	NA	0	Enable OTP readback protection  <b>Note:</b> WEN must be set.

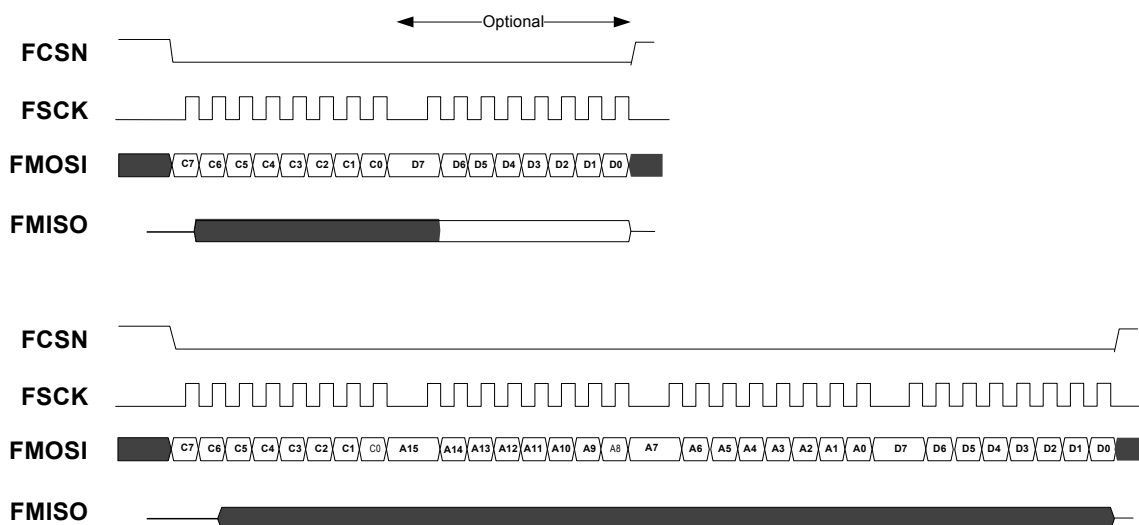
**Table 34** OTP operation commands

The signaling of the SPI interface is shown in **Figure 31** and **Figure 32**



**Figure 31** SPI read operation for direct and addressed command

**Note:** For the READ command there must be a delay,  $T_{Srd}$ , from the last address bit to the first data bit. Minimum value of  $T_{Srd}$  is 4 clock cycles (**xc1**) + 10ns, which is 260 ns when **xc1** is 16 MHz.



**Figure 32** SPI write operations for direct and addressed commands

Abbreviations	Description
Cx	SPI Command bit
Ax	OTP address. Sequence MS to LS byte, MS to LS bit.
Dx	SPI data bit, Sequence LS to MS byte, MS to LS bit. Presence depending on SPI command.

**Table 35** OTP SPI interface signal abbreviations

#### WREN / WRDIS OTP write enable/disable:

SPI commands WREN and WRDIS sets and resets the OTP write enable latch WEN in register FSR. This latch enables all write operations in the OTP blocks.

The device will power-up in write disable state, and automatically go back to write disable state after each write SPI command (FCSN set high). Each write command over the SPI interface must therefore be preceded by a WREN command.

Both WREN and WRDIS are 1-byte SPI commands with no data.

#### RDSR / WRSR read/write OTP status register

SPI commands RDSR and WRSR read and writes to the OTP status register FSR. Both commands are 1 and are followed by a data byte for the FSR content, see [chapter 16 on page 126](#).

#### READ

SPI command READ reads out the content of an addressed position in the OTP main block. It must be followed by 2 bytes denoting the start address of the read operation, see [chapter 16 on page 126](#). If bit INFEN in register FSR is enabled, the read operation will be conducted from System Space instead.

If the FCSN line is kept active after the first data byte is read out the read command can be extended, the address is auto incremented and data continues to shift out. The internal address counter rolls over when the highest address is reached, allowing the complete memory to be read in one continuous read command.

A read back of the OTP main block content is only possible if the read disable bit **RDIS** in the FSR register is not set.

## PROGRAM

SPI command PROGRAM, programs the content of the addressed position in the OTP main block. It must be followed by 2 bytes denoting the start address of the write operation, see *chapter 16 on page 126*. If bit INFEN in register FSR is enabled, the write operation will access the System Space instead.

Before each write operation the write enable latch WEN must be enabled through the WREN SPI command. It is possible to write up to 1 kB in one PROGRAM command. The first byte can be at any address.

The device automatically returns to OTP write disable (WEN=0) after completion of a PROGRAM command (pin FCSN=1).

## RDISMB - Enable Read DISable of MainBlock

SPI command RDISMB enables the readback protection of the OTP. The command disables all read and write access to the OTP main block from the SPI interface. It also disables write operations in the InfoPage, but read InfoPage read operations are still possible. This will protect code and data in the device from being retrieved through the SPI interface.

Before the RDISMB command the write enable latch WEN must be enabled through the WREN SPI command. Once the RDISMB command is sent all SPI connection/control of the OTP from the SPI interface is lost. It is important that this command is the last one to be sent in a OTP programming sequence.

The command is a 1 byte command with no data.

## 7 Random Access memory (RAM)

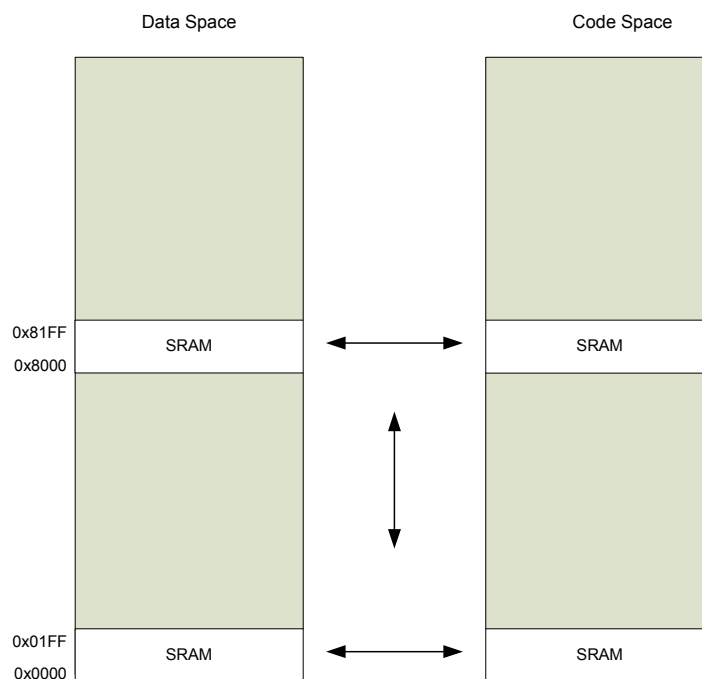
There are two separate RAM blocks. These blocks are used to save temporary data or programs.

The MCU internal RAM (IRAM) is the fastest and most flexible, but with only 256 bytes it is very limited.

To accommodate more temporary storage of data or code there is an additional 512x8bit (512 bytes) SRAM memory block default located in the XDATA address space from address 0x0000 to 0x01FF. The location of the SRAM blocks in the MCU address space can be changed, see [section 7.1](#).

### 7.1 SRAM configuration

It is possible to configure the location in address space of the SRAM block as described in [Figure 33](#)



**Figure 33** Configurability of SRAM address space location

You can address the SRAM memory blocks both as data and code. The MEMCON register controls this behavior:

Addr	Bit	R/W	Function	Reset value: 0x00
0xA7	7:3	-	Reserved	
	2	R/W	SRAM address location: 0: SRAM blocks start from address 0x0000 1: SRAM blocks start from address 0x8000	
	1	-	Reserved	
	0	R/W	SRAM mapping: 0: Mapped as data 1: Mapped as code	

**Table 36** MEMCON register

## 8 Timers/counters

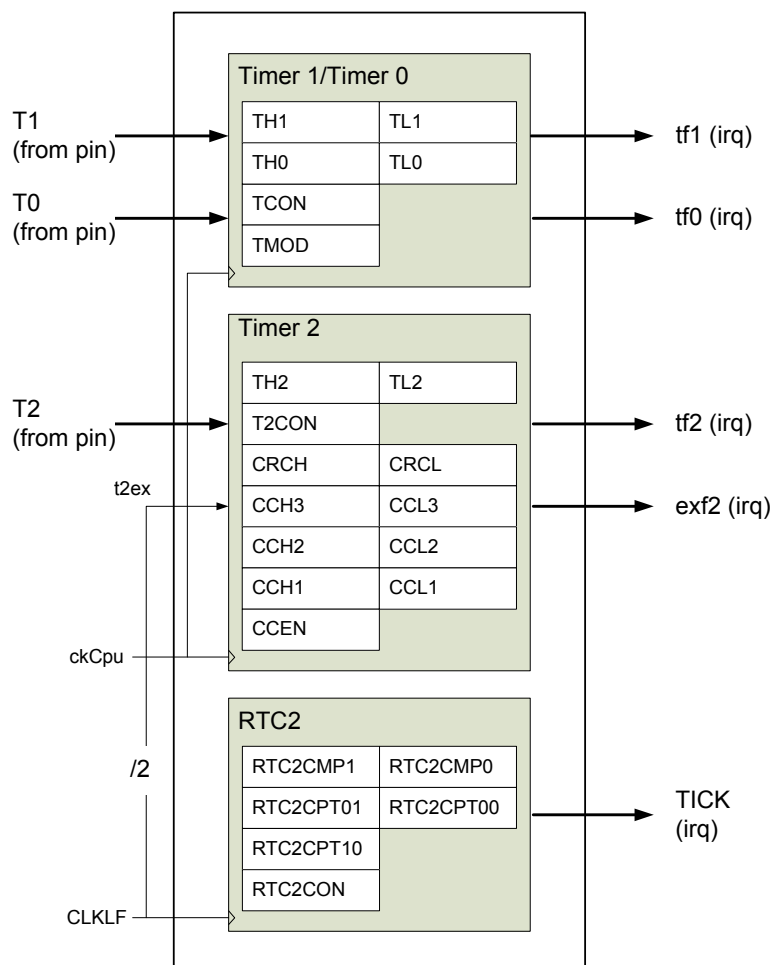
There are a set of counters used for timing up important system events. One of the timers (RTC2) is also available in power down mode where it can be used as a wakeup source.

### 8.1 Features

The following set of timers/counters are included:

- Three 16-bit timers/counters (Timer 0, Timer 1 and Timer 2) which can operate as either a timer with a clock rate based on the MCU clock, or as an event counter clocked by signals from the programmable digital I/O.
- RTC2 is a configurable, linear, 16-bit real time clock with capture and compare capabilities. Input clock frequency is 32.768 KHz.

### 8.2 Block diagram



**Figure 34** Block diagram of timers/counters



## 8.3 Functional description

### 8.3.1 Timer 0 and Timer 1

In timer mode, Timers 0 and 1 are incremented every 12 clock cycles.

In the counter mode, the Timers 0 and 1 are incremented when the falling edge is detected at the corresponding input pin T0 for Timer 0, or T1 for Timer 1.

**Note:** Timer input pins **T0**, **T1** and, **T2** must be configured as described in [section 8.4 on page 85](#).

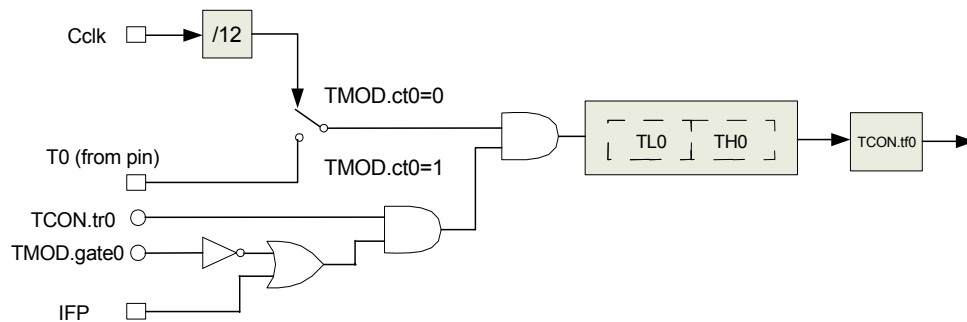
Since it takes two clock cycles to recognize a 1-to-0 event, the maximum input count rate is  $\frac{1}{2}$  of the oscillator frequency. There are no restrictions on the duty cycle, however to ensure proper recognition of 0 or 1 state, an input should be stable for at least 1 clock cycle.

Timer 0 and Timer 1 status and control are in TCON and TMOD register. The actual 16-bit Timer 0 value is in TH0 (8 msb) and TL0 (8 lsb), while Timer 1 uses TH1 and TL1.

Four operating modes can be selected for Timers 0 and 1. Two Special Function Registers, TMOD and TCON, are used to select the appropriate mode.

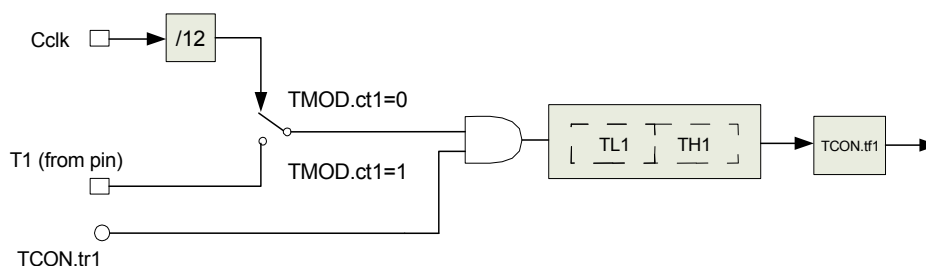
#### 8.3.1.1 Mode 0 and Mode 1

In mode 0, Timers 0 and 1 are each configured as a 13-bit register (TL0/TL1 = 5 bits, TH0/TH1 = 8 bits). The upper three bits of TL0 and TL1 are unchanged and should be ignored. In mode 1 Timer 0 is configured as a 16-bit register.



**Figure 35** Timer 0 in mode 0 and 1

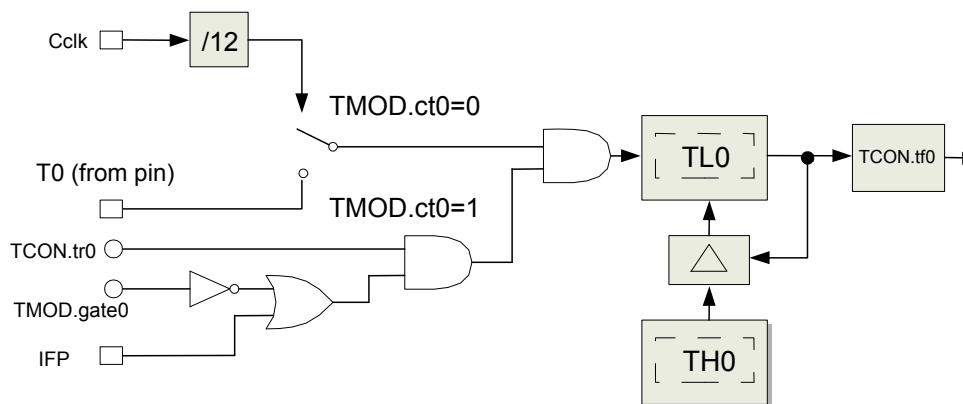
Likewise, in mode 1, Timer 1 is configured as a 16-bit register.



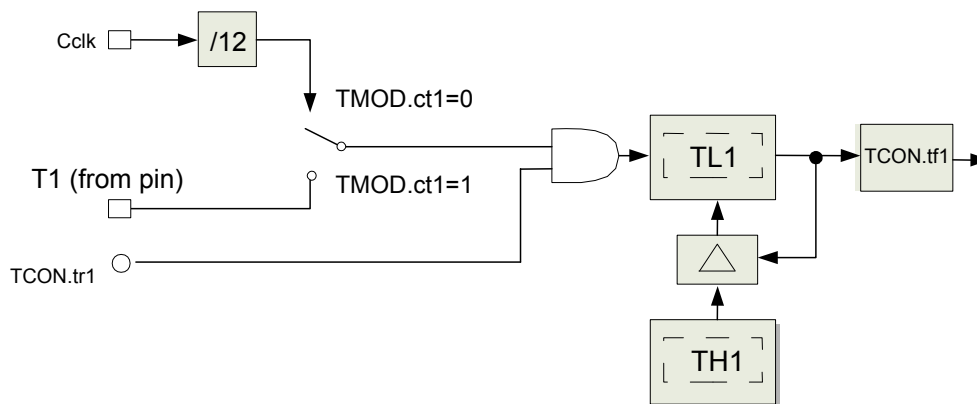
**Figure 36** Timer 1 in mode 0 and 1

### 8.3.1.2 Mode 2

In this mode, Timers 0 and 1 are each configured as an 8-bit register with auto reload.



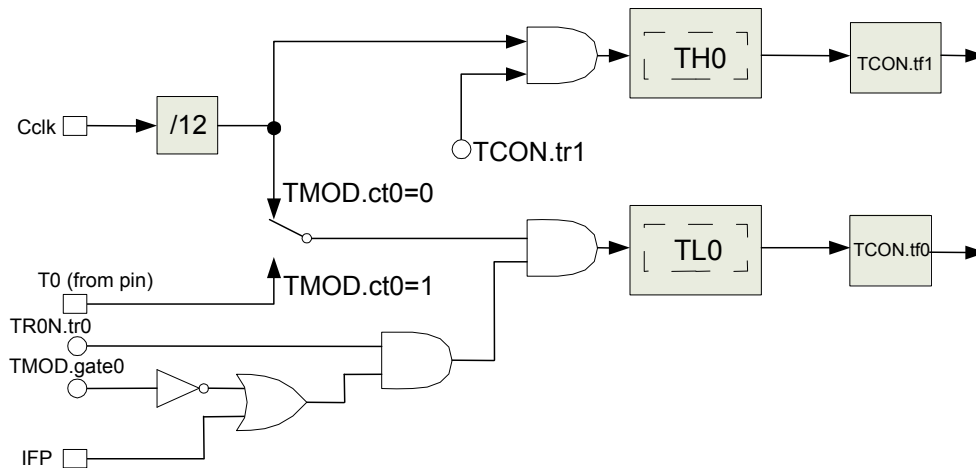
**Figure 37** Timer 0 in mode 2



**Figure 38** Timer 1 in mode 2

### 8.3.1.3 Mode 3

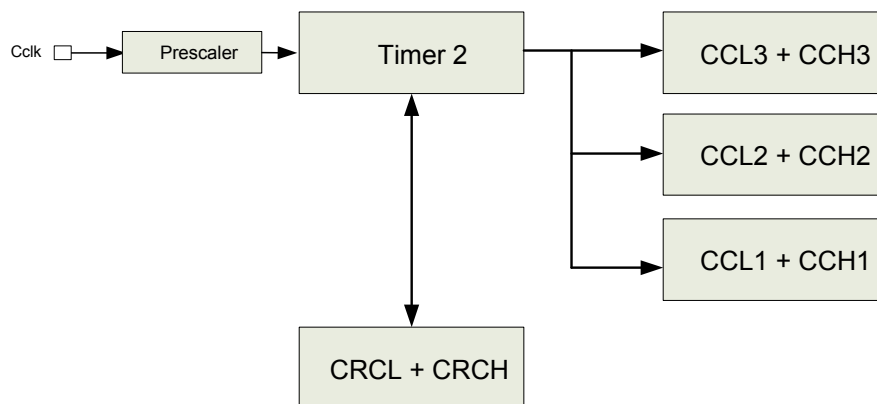
In mode 3 Timers 0 and 1 are configured as one 8-bit timer/counter and one 8-bit timer, but timer 1 in this mode holds its count. When Timer 0 works in mode 3, Timer 1 can still be used in other modes by the serial port as a baud rate generator, or as an application not requiring an interrupt from Timer 1.



**Figure 39** Timer 0 in mode 3

### 8.3.2 Timer 2

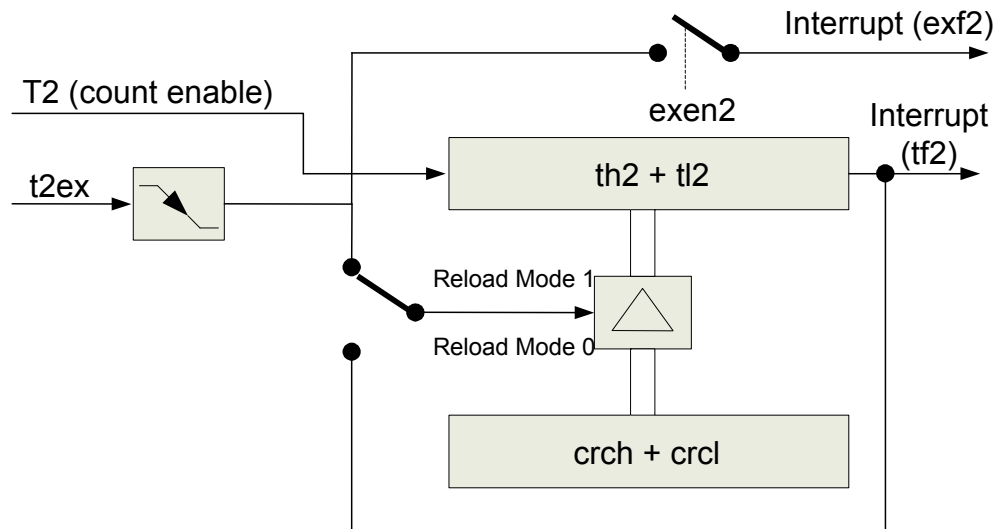
Timer 2 is controlled by T2CON while the value is in TH2 and TL2. Timer 2 also has four capture and one compare/reload registers which can read a value without pausing or reload a new 16-bit value when Timer 2 reaches zero, see [chapter 8.4.7 on page 88](#) and [chapter 8.4.8 on page 88](#).



**Figure 40** Timer 2 block diagram

### 8.3.2.1 Timer 2 description

Timer 2 can operate as a timer, event counter, or gated timer.



**Figure 41** Timer 2 in Reload Mode

### 8.3.2.2 Timer mode

Timer mode is invoked by setting the  $t2i0=1$  and  $t2i1=0$  in the T2CON register. In this mode, the count rate is derived from the clk input.

Timer 2 is incremented every 12 or 24 clock cycles depending on the 2:1 prescaler. The prescaler mode is selected by bit  $t2ps$  of T2CON register. When  $t2ps=0$ , the timer counts up every 12 clock cycles, otherwise every 24 cycles.

### 8.3.2.3 Event counter mode

This mode is invoked by setting the  $t2i0=0$  and  $t2i1=1$  in the T2CON register.

In this mode, Timer 2 is incremented when external signal T2 (see [section 8.4 on page 85](#) for more information on T2) changes its value from 1 to 0. The T2 input is sampled at every rising edge of the clock. Timer 2 is incremented in the cycle following the one in which the transition was detected. The maximum count rate is  $\frac{1}{2}$  of the clock frequency.

### 8.3.2.4 Gated timer mode

This mode is invoked by setting the  $t2i0=1$  and  $t2i1=1$  in the T2CON register.

In this mode, Timer 2 is incremented every 12 or 24 clock cycles (depending on T2CON  $t2ps$  flag). Additionally, it is gated by the external signal T2. When  $T2=0$ , Timer 2 is stopped.

### 8.3.2.5 Timer 2 reload

A 16-bit reload from the CRC register can be done in two modes:

- Reload Mode 0: Reload signal is generated by Timer 2 overflow (auto reload).
- Reload Mode 1: Reload signal is generated by negative transition at t2ex.

**Note:** t2ex is connected to an internal clock signal which is half frequency of CLKLF (see *section 11 on page 99.*)

## 8.4 SFR registers

### 8.4.1 Timer/Counter control register – TCON

TCON register reflects the current status of MCU Timer 0 and Timer 1 and it is used to control the operation of these modules.

Address	Reset value	Bit	Name	Auto clear	Description
0x88	0x00	7	tf1	Yes	Timer 1 overflow flag. Set by hardware when Timer1 overflows.
		6	tr1	No	Timer 1 Run control. If cleared, Timer 1 stops.
		5	tf0	Yes	Timer 0 overflow flag. Set by hardware when Timer 0 overflows.
		4	tr0	No	Timer 0 Run control. If cleared, Timer 0 stops.
		3	ie1	Yes	External interrupt 1 flag. Set by hardware.
		2	it1	No	External interrupt 1 type control. 1: falling edge, 0: low level
		1	ie0	Yes	External interrupt 0 flag. Set by hardware.
		0	it0	No	External interrupt 0 type control. 1: falling edge, 0: low level

**Table 37** TCON register

The tf0, tf1 (Timer 0 and Timer 1 overflow flags), ie0 and ie1 (external interrupt 0 and 1 flags) are automatically cleared by hardware when the corresponding service routine is called.

## 8.4.2 Timer mode register - TMOD

TMOD register is used for configuration of Timer 0 and Timer 1.

Address	Reset value	Bit	Name	Description
0x89	0x00	7	gate1	Timer 1 gate control
		6	ct1	Timer 1 counter/timer select. 1: Counter, 0: Timer
		5:4	mode1	Timer 1 mode 00 – Mode 0: 13-bit counter/timer 01 – Mode 1: 16-bit counter/timer 10 – Mode 2: 8-bit auto-reload timer 11 – Mode 3: Timer 1 stopped
		3	gate0	Timer 0 gate control
		2	ct0	Timer 0 counter/timer select. 1: Counter, 0: Timer
		1:0	mode0	Timer 0 mode 00 – Mode 0: 13-bit counter/timer 01 – Mode 1: 16-bit counter/timer 10 – Mode 2: 8-bit auto-reload timer 11 – Mode 3: two 8-bit timers/counters

*Table 38 TMOD register*

## 8.4.3 Timer 0 – TH0, TL0

Address	Register name
0x8A	TL0
0x8C	TH0

*Table 39 Timer 0 register (TH0:TL0)*

These registers reflect the state of Timer 0. TH0 holds higher byte and TL0 holds lower byte. Timer 0 can be configured to operate as either a timer or a counter.

## 8.4.4 Timer 1 – TH1, TL1

Address	Register name
0x8B	TL1
0x8D	TH1

*Table 40 Timer 1 register (TH1:TL1)*

These registers reflect the state of Timer 1. TH1 holds higher byte and TL1 holds lower byte. Timer 1 can be configured to operate as either timer or counter.

### 8.4.5 Timer 2 control register – T2CON

T2CON register reflects the current status of Timer 2 and is used to control the Timer 2 operation.

Address	Reset value	Bit	Name	Description
0xC8	0x00	7	t2ps	Prescaler select. 0: timer 2 is clocked with 1/12 of the ckCpu frequency. 1: timer 2 is clocked with 1/24 of the ckCpu frequency.
		6	i3fr	Int3 edge select. 0: falling edge, 1: rising edge
		5	i2fr	Int2 edge select. 0: falling edge, 1: rising edge
		4:3	t2r	Timer 2 reload mode. 0X – reload disabled, 10 – Mode 0, 11 – Mode 1
		2	t2cm	Timer 2 compare mode. 0: Mode 0, 1: Mode 1
		1:0	t2i	Timer 2 input select. 00: stopped, 01: f/12 or f/24, 10: falling edge of T2, 11: f/12 or f/24 gated by T2.

**Table 41** T2CON register

### 8.4.6 Timer 2 – TH2, TL2

Address	Register name
0xCC	TL2
0xCD	TH2

**Table 42** Timer 2 (TH2:TL2)

The TL2 and TH2 registers reflect the state of Timer 2. TH2 holds higher byte and TL2 holds lower byte. Timer 2 can be configured to operate in compare, capture or, reload modes.

### 8.4.7 Compare/Capture enable register – CCEN

The CCEN register serves as a configuration register for the Compare/Capture Unit associated with the Timer 2.

Address	Reset value	Bit	Name	Description
0xC1	0x00	7:6	coca3	compare/capture mode for CC3 register 00: compare/capture disabled 01: reserved 10: reserved 11: capture on write operation into register CCL3
		5:4	coca2	compare/capture mode for CC2 register 00: compare/capture disabled 01: reserved 10: reserved 11: capture on write operation into register CCL2
		3:2	coca1	compare/capture mode for CC1 register 00: compare/capture disabled 01: reserved 10: reserved 11: capture on write operation into register CCL1
		1:0	coca0	compare/capture mode for CC0 register 00: compare/capture disabled 01: reserved 10: compare enabled 11: capture on write operation into register CCL0

**Table 43** CCEN register

### 8.4.8 Capture registers – CC1, CC2, CC3

The Compare/Capture registers (CC1, CC2, CC3) are 16-bit registers used by the Compare/Capture Unit associated with the Timer 2. CCHn holds higher byte and CCLn holds lower byte of the CCn register.

Address	Register name
0xC2	CCL1
0xC3	CCH1
0xC4	CCL2
0xC5	CCH2
0xC6	CCL3
0xC7	CCH3

**Table 44** Capture Registers - CC1, CC2 and CC3



## 8.4.9 Compare/Reload/Capture register – CRCH, CRCL

Address	Reset value	Register name
0xCA	0x00	CRCL
0xCB	0x00	CRCH

**Table 45** Compare/Reload/Capture register - CRCH, CRCL

CRC (Compare/Reload/Capture) register is a 16-bit wide register used by the Compare/Capture Unit associated with Timer 2. CRCH holds higher byte and CRCL holds lower byte.

## 8.5 Real Time Clock - RTC

RTC2 contains two registers that can be used for capturing timer values; one loaded at positive edge of the 32.768 kHz clock and another register clocked by the MCU clock for better resolution. Both registers are updated as a consequence of an external event. RTC2 can also give an interrupt at predefined intervals due to value equality between the timer and a compare register. RTC2 ensures that the functions the interrupt is used for are awoken prior to the interrupt.

### 8.5.1 Features

- 32.768 kHz, sub- $\mu$ A.
- 16-bit.
- Linear.
- Compare with IRQ (TICK). Resolution: 30.52  $\mu$ s.
- Capture with increased resolution: 125 ns.

### 8.5.2 Functional description of SFR registers

The following registers control RTC2.

Address (Hex)	Name/Mnemonic	Bit	Reset value	Type	Description
0xB3	RTC2CON	4:0		R/W	RTC2 configuration register.
	<i>sfrCapture</i>	4	0	W	Trigger signal. When the MCU writes a '1' to this register field, RTC2 will capture the timer value. The value is stored in RTC2CPT00 and RTC2CPT01. An additional counter clocked by the MCU clock will at this point contain the number of MCU clock cycles from the previous positive edge of the 32.768 kHz clock (edge detect @ MCU clock). The value is stored in RTC2CPT1.
	<i>enableExternalCapture</i>	3	0	R/W	<b>1:</b> Timer value is captured if required by an IRQ from the Radio (edge detect @ MCU clock). The value is stored in RTC2CPT00 and RTC2CPT01. An additional counter clocked by the MCU clock will at this point contain the number of MCU clock cycles from the previous positive edge of the 32.768 kHz clock (edge detect @ MCU clock). The value is stored in RTC2CPT1. <b>0:</b> Capture by Radio disabled.

Address (Hex)	Name/Mnemonic	Bit	Reset value	Type	Description
	<i>compareMode</i>	2:1	00	R/W	Compare mode. <b>11:</b> The Rtc2 IRQ is assigned when the timer value is equal to the concatenation of RTC2CMP1 and RTC2CMP0. RTC2 ensures that the functions for which the IRQ is intended, are all awoken prior to the Rtc2 IRQ. When the Rtc2 IRQ is assigned, the timer is reset. <b>10:</b> Same as above, except that the Rtc2 IRQ will <i>not</i> reset the timer. The timer will always wrap around at overflow. <b>0x:</b> Compare disabled. OK
	<i>rtc2Enable</i>	0	0	R/W	<b>1:</b> RTC2 is enabled. The clock to the RTC2 core functionality is running. <b>0:</b> RTC2 is disabled. The clock to the RTC2 core functionality stands still and the timer is reset.
0xB4	RTC2CMP0	7:0	0xFF	R/W	RTC2 compare value register 0. Contains LSByte of the value to be compared to the timer value to generate Rtc2 IRQ. Resolution: 30.52 µs.
0xB5	RTC2CMP1	7:0	0xFF	R/W	RTC2 compare value register 1. Contains MSByte of the value to be compared to the timer value to generate Rtc2 IRQ.
0xB6	RTC2CPT00	7:0	0x00	R	RTC2 capture value register 00. Contains LSByte of the timer value at the time of the capture event. Resolution: 30.52 µs.
0xAB	RTC2CPT01	7:0	0x00	R	RTC2 capture value register 01. Contains MSByte of the timer value at the time of the capture event.
0xAC	RTC2CPT10	7:0	0x00	R	RTC2 capture value register 1. Contains the value of the counter that counts the number of MCU clock cycles from the previous positive edge of the 32.768 kHz clock until the capture event. The counter value is truncated by one bit (LSBit). Resolution: 125 ns.
0xE3	RTC2PC0	6:0	0x63	R/W	RTC2 Pre-start time. Contains the RTC2 pre-start time in CLKF periods. The reset value is 63; however, if another value is needed, you must write the value to this register at the start of your program. Note only the 7 LSBit of the register must be changed, the MSByte is reserved.

**Table 46** RTC2 register map

The Rtc2 timer is a 16 bit timer counting from zero and upwards at the rate of the 32.768 kHz clock. When the Rtc2 timer is equal to the concatenation of RTC2CMP1 and RTC2CMP0, an Rtc2 IRQ, also referred to as TICK, is generated. There is an uncertainty of one CLKLF period, 30.52 µs, from when the Rtc2 is started or a new value is given to the RTC2 compare value registers and until the IRQ is given.

The time for the IRQ is given by the range:

$$\left[ \frac{[\text{RTC2CMP1} : \text{RTC2CMP0}] - \text{timer}}{32768}, \frac{[\text{RTC2CMP1} : \text{RTC2CMP0}] - \text{timer} + 1}{32768} \right] [\text{s}]$$

where [RTC2CMP1:RTC2CMP0] is the concatenation of RTC2CMP1 and RTC2CMP0 into a 16 bits word and **timer** is the current value of the Rtc2 timer when the RTC2 compare value register was updated or the Rtc2 enabled.

If compare mode 11 is used, the Rtc2 IRQ will be given every second.

$$\frac{[\text{RTC2CMP1: RTC2CMP0}] + 1}{32768} [\text{s}]$$

The RTC2 compare value is updated every time RTC2CMP1 or RTC2CMP0 is written. This might give unwanted behavior if precaution is not taken when updating any of the variables. When new values are written to RTC2CMP1 and RTC2CMP0, the Rtc2 IRQ should be disabled to prevent unwanted Rtc2 IRQ.

To make sure everything is up and running when the Rtc2 IRQ is given in Register retention, the MCU is pre-started before the IRQ is given. If XOSC16M is enabled, the pre-start time is long enough to make sure that this clock is up and running before the IRQ is given<sup>1</sup>. If RCOSC16M is enabled by CLKCTRL[5:4], this will be the clock source in the pre-start period. To save power, the user could choose to go to Standby while waiting for the IRQ. If only RCOSC16M is enabled, the pre-start time is shorter, making sure that the RC-oscillator is up and running before the Rtc2 IRQ is given. This same, short pre-startup time is used from Register Retention to Active if XOSC16M is running while in Register retention<sup>2</sup>CLKCTRL[7] = 1.

This implies that the time from going to Register retention and until the Rtc2 IRQ is given, must always be longer then the pre-start time: 63 CLKLF periods for the long pre-start, or you can override this value by setting register RTC2PCO, and 2 CLKLF for the short pre-start.

The Rtc2 counter uses the 32.768 kHz low frequency clock for the Rtc2 timer, and one of the 32.768 kHz sources must be enabled when using the Rtc2. See **section 13.3 on page 112** for the 32.768 kHz clock.

Reading RTC2CMP0 and RTC2CMP1:

- Disable the Rtc2 IRQ, until both registers have been written.

Reading RTC2CPT00, RTC2CPT01 and RTC2CPT10:

- Disable The Radio IRQ until all three registers have been read.

Uncertainty in capture values:

- 250 ns.

- 
1. The crystal start-up time must be <2 ms to ensure that XOSC16M is clock source on arrival of the Rtc2 IRQ. Refer to **section 13.3.1 on page 112** for further details.
  2. To get the short pre-startup time when going to Register retention with XOSC16M running in the power down mode, make sure XOSC16M is running before going to Register retention. If it is not, the long pre-start time is used, and the minimum value for the long pre-startup for the RTC2 compare value register should be used. This applies only the first time going to Register retention after enabling XOSC16M in Register retention.

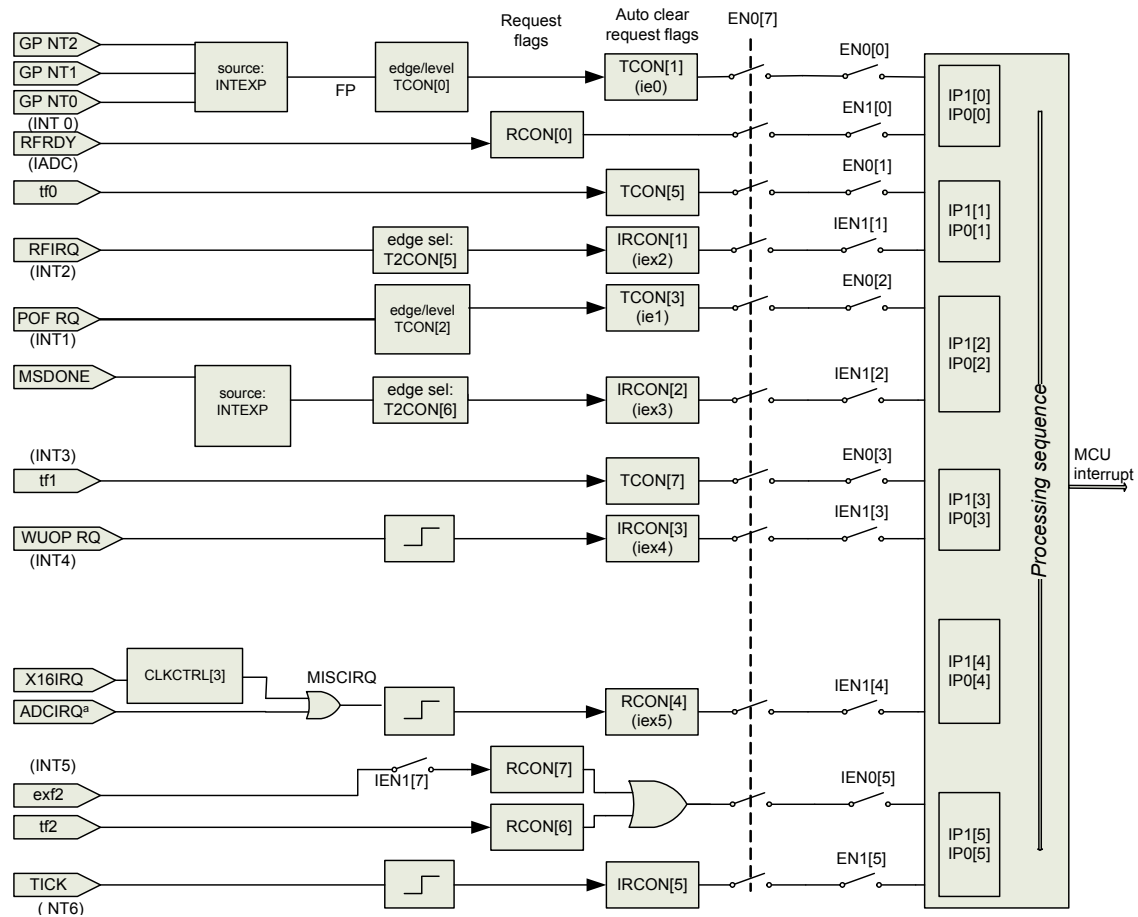
## 9 Interrupts

nRF31512 has 13 interrupt sources, as shown in **Figure 42**. The unit manages dynamic program sequencing based upon important real-time events as signalled from timers, the RF transceiver, pin activity, and so on.

### 9.1 Features

- Interrupt controller with 13 sources and 4 priority levels
- Interrupt request flags available
- Interrupt from pin (configurable)

### 9.2 Block diagram



a. ADCIRQ: for nRF31512 only.

**Figure 42** Block diagram of interrupt structure

## 9.3 Functional description

When an enabled interrupt occurs, the MCU vectors to the address of the interrupt service routine (ISR) associated with that interrupt, as listed in **Table 47**. The MCU executes the ISR to completion unless another interrupt of higher priority occurs.

Source	vector	Polarity	Description
IFP	0x0003	low/fall	Interrupt from pin <b>GP INT0</b> , <b>GP INT1</b> or <b>GP INT2</b> as selected by bits 3,4 or 5 in SFR INTEXP. Only one of the bits may be set at a time.
tf0	0x000B	high	Timer 0 overflow interrupt
POFIRQ	0x0013	low/fall	Power Failure interrupt
tf1	0x001B	high	Timer 1 overflow interrupt
tf2	0x002B	high	Timer 2 overflow interrupt
exf2	0x002B	high	Timer 2 external reload
RFRDY	0x0043	high	RF SPI ready
RFIRQ	0x004B	fall/rise	RF interrupt
MSDONE	0x0053	fall/rise	Master SPI transaction completed
WUOPIRQ	0x005B	fall <sup>1</sup>	Wakeup on pin interrupt
MISCIRQ	0x0063	rise	Miscellaneous interrupt is the sum of: <ul style="list-style-type: none"> <li>• XOSC16M started (X16IRQ)</li> <li>• ADC Ready (ADCIRQ) interrupt<sup>2</sup></li> </ul>
TICK	0x006B	rise	Internal Wakeup (from RTC2) interrupt

1. Polarity is always fall on interrupt, but pin polarity can be selected by bit 2 in the OPMCON register as described in **Table 61 on page 106**.
2. Only for nRF31512.

**Table 47** Interrupt sources

**Note:** RFIRQ, WUOPIRQ, MISCIRQ and TICK are not activated unless wakeup is enabled by WUCON (see **section 11.3.5 on page 107**).

## 9.4 SFR registers

Various SFR registers are used to control and prioritize between different interrupts.

The TCON, IRCON, SCON, IP0, IP1, IEN0, IEN1 and INTEXP are described in this section. In addition the TCON and T2CON are used, the description for these registers can be found in **chapter 8 on page 80**.

### 9.4.1 Interrupt Enable 0 Register – IEN0

The IEN0 register is responsible for global interrupt system enabling/disabling and also Timer 0, 1 and 2, Port 0 and Serial Port individual interrupts enabling/disabling.

Address	Bit	Description
0xA8	7	1: Enable interrupts. 0: <b>all</b> interrupts are disabled
	6	Not used
	5	1: Enable Timer2 (tf2/exf2) interrupt.
	4	Not used
	3	1: Enable Timer1 overflow (tf1) interrupt
	2	1: Enable Power failure (POFIRQ) interrupt
	1	1: Enable Timer0 overflow (tf0) interrupt.
	0	1: Enable Interrupt From Pin (IFP) interrupt.

*Table 48 IEN0 register*

### 9.4.2 Interrupt Enable 1 Register – IEN1

The IEN1 register is responsible for RF, SPI and Timer 2 interrupts.

Address	Bit	Description
0xB8	7	1: Enable Timer2 external reload (exf2) interrupt
	6	Not used
	5	1: Internal wakeup (TICK) interrupt enable
	4	1: Miscellaneous (MISCIRQ) interrupt enable
	3	1: Wakeup on pin (WUOPIRQ) interrupt enable
	2	1: SPI master completed (MSDONE) interrupt enable
	1	1: RF (RFIRQ) interrupt enable
	0	1: RF SPI ready (RFRDY) interrupt enable

*Table 49 IEN1 register*

Address	Bit	Description	Reset value 0x01
0xA6	7:6	Not used	
	5	1: Enable GP INT2 (from pin) to IFP	
	4	1: Enable GP INT1 (from pin) to IFP	
	3	1: Enable GP INTO (from pin) to IFP	
	2	Not used	
	1	1: Enable Master SPI completed (MSDONE)interrupt	
	0	Not used	

*Table 50 INTEXP register*

### 9.4.3 Interrupt Priority Registers – IP0, IP1

The interrupt sources are grouped into six priority groups. For each of the groups, one of four priority levels can be selected. They can be selected by setting appropriate values in IP0 and IP1 registers.

The contents of the Interrupt Priority registers define the priority levels for each interrupt source according to the tables below.

Address	Bit	Description
0xA9	7:6	Not used
	5:0	Interrupt priority. Each bit together with corresponding bit from IP1 register specifies the priority level of the respective interrupt priority group.

*Table 51 IP0 register*

Address	Bit	Description
0xB9	7:6	Not used
	5:0	Interrupt priority. Each bit together with corresponding bit from IP0 register specifies the priority level of the respective interrupt priority group.

*Table 52 IP1 register*

Group	Interrupt bits		Priority groups
0	IP1[0], IP0[0]	IFP	RFRDY
1	IP1[1], IP0[1]	tf0	RFIRQ
2	IP1[2], IP0[2]	POFIRQ	MSDONE
3	IP1[3], IP0[3]	tf1	WUOPIRQ
4	IP1[4], IP0[4]	MISCIRQ	
5	IP1[5], IP0[5]	tf2/exf2	TICK

*Table 53 Priority groups*

IP1.x	IP0.x	Priority level
0	0	Level 0 (lowest)
0	1	Level 1
1	0	Level 2
1	1	Level 3 (highest)

*Table 54 Priority levels (x is the number of priority group)*

## 9.4.4 Interrupt Request Control Registers – IRCON

The IRCON register contains interrupt request flags.

Address	Bit	Auto clear	Description
0xC0	7	No	Timer 2 external reload (exf2) interrupt flag
	6	No	Timer 2 overflow (tf2) interrupt flag
	5	Yes	Internal wakeup (TICK) interrupt flag
	4	Yes	Miscellaneous (MISCIRQ) interrupt flag
	3	Yes	Wakeup on pin (WUOPIRQ) interrupt flag
	2	Yes	Master SPI (MSDONE) interrupt flag
	1	Yes	RF (RFIRQ) interrupt flag
	0	No	RF SPI ready (RFRDY) interrupt flag

**Table 55** IRCON register



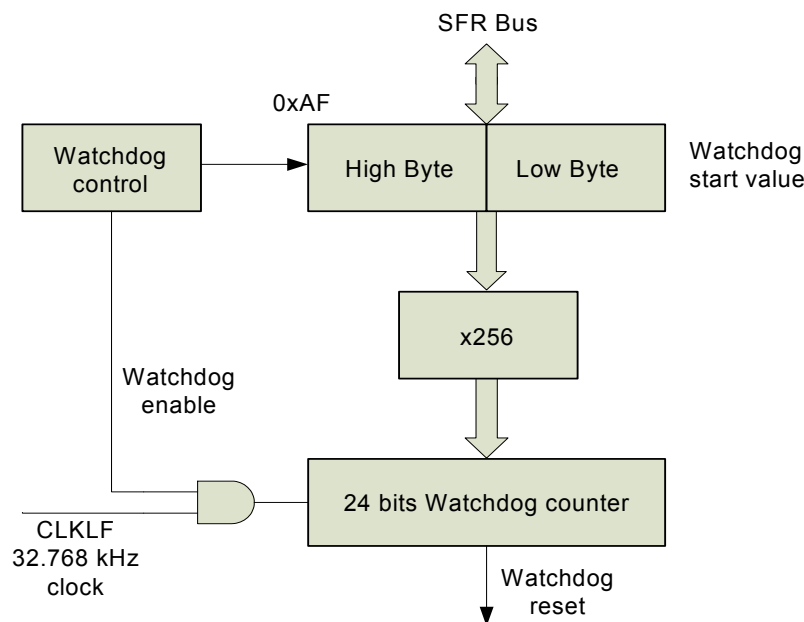
## 10 Watchdog

The on-chip watchdog forces a system reset if the running software for some reason encounters a hang situation.

### 10.1 Features

- 32.768 kHz, sub- $\mu$ A.
- 16-bit with an offset of 8 bits.
- Minimum Watchdog timeout interval: 7.8125 ms.
- Maximum Watchdog timeout interval: 512 s.
- Disable (reset) only by a system reset, or possibly when the chip enters the following power saving modes: Register retention. See [section 15.3.1 on page 121](#) for details.

### 10.2 Block diagram



**Figure 43** Watchdog block diagram

### 10.3 Functional description

The following register controls the Watchdog.

Address (Hex)	Name/Mnemonic	Bit	Reset value	Type	Description
0xAF	WDSV	15:0	0x0000	R/W	Watchdog start value register. MSByte and LSByte of the word are written and read as separate bytes.

**Table 56** Watchdog register

watchdogStartValue (WDSV) contains the upper 16 bits of the Watchdog counter's initial value. This 16 bits word is read and written as two separate bytes, LSByte and MSByte. LSByte is read and written first. After a write to WDSV, the next read of WDSV will always give the LSByte, and after a read, the next byte written will always be to the LSByte. In other words, to write to WDSV, two bytes must be written without a read between the writes, and vice-versa for read operations. Readout of WDSV will not give the current value of the Watchdog counter, but the start value for the counter.

After a reset, the default state of the Watchdog is disabled. The Watchdog is activated when both bytes of WDSV have been written. The Watchdog counter then counts down from WDSV\*256 towards 0. When 0 is reached, the complete microcontroller, as well as the peripherals, are reset. A reset from the Watchdog will have the same effect as a power-on reset or a reset from pin. To avoid the reset, the software must reload WDSV sufficiently often. The Watchdog counter is updated with a new start value and restarted every time WDSV is written.

The Watchdog counter uses the 32.768 kHz low frequency clock, and one of the 32.768 kHz sources must be enabled when using the Watchdog. See **section 13.3 on page 112** for the 32.768 kHz clock.

The Watchdog timeout is given by:

$$\frac{\text{WDSV} \times 256}{32768} [\text{s}]$$

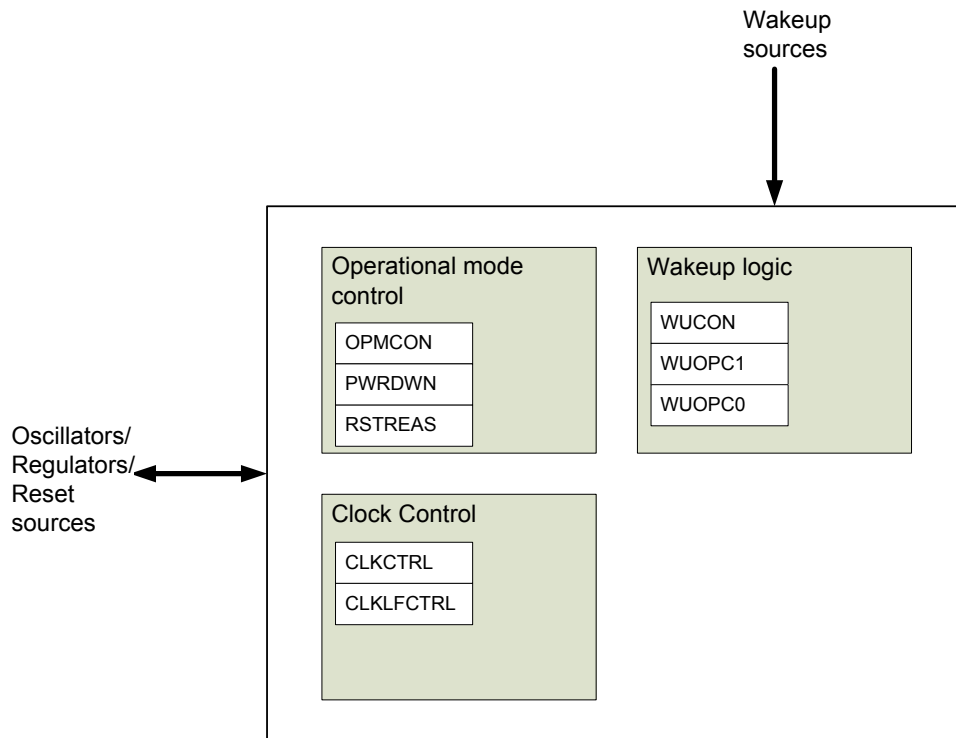
If WDSV is loaded with 0x0000, the maximum Watchdog timeout interval of 512 seconds is used, i.e. the Watchdog is not disabled.

If the Watchdog has been started, it can only be disabled (reset) by a system reset, or possibly when the chip enters the Register retention power-saving mode. Please refer to OPMCON bit 0 in **Table 61 on page 106**.

## 11 Power and clock management

The Power Management function controls the power dissipation through administration of modes of operation and by controlling clock frequencies.

### 11.1 Block diagram



**Figure 44** Block diagram of power and clock management

### 11.2 Modes of operation

After the device is reset or powered on it enters active mode and the functional behavior is controlled by software. To enter one of the power saving modes, the PWRDWN register must be written with selected mode (as data).

To re-enter the active mode a wakeup source (valid for given power down mode) has to be activated.

The modes of operation are summarized in the following table:

Mode	Brief description
Register retention, timers off <sup>1</sup>	<p>Current: See <i>Table 88 on page 142</i></p> <p>Powered functions:</p> <ul style="list-style-type: none"> <li>• Pins inclusive wakeup filter</li> <li>• Power Manager</li> <li>• IRAM and SRAM</li> <li>• RCOSC32K</li> <li>• RTC2 and watchdog clocked on 32 KHz clock</li> <li>• All registers</li> <li>• Optional: XOSC16M</li> </ul> <p>Wakeup source(s): From pin. See also footnote for bit 7 regarding wakeup in same table.</p> <p>Start-up time: Wake-up from pin:</p> <ul style="list-style-type: none"> <li>• &lt; 100 µs when starting on RCOSC16M</li> </ul> <p>Comment: Wakeup does not lead to system reset (after wakeup, program execution will resume from the current instruction).</p>

Mode	Brief description
Register retention, timers on <sup>1</sup> .	<p>Current: See <b>Table 88 on page 142</b></p> <p>Powered functions: In addition to Register retention, timers off:</p> <p>Wakeup source(s): From pin or wakeup TICK from timer.</p> <p>Start-up time: Wakeup from pin:</p> <ul style="list-style-type: none"> <li>&lt; 100 µs when starting on RCOSC16M</li> </ul> <p>Wakeup TICK:</p> <ul style="list-style-type: none"> <li>Pre-start voltage regulators and XOSC16M, system ready on RTC2 TICK. To save power, the user may choose to enter Standby power-down mode when the MCU system is awoken (&lt;100 µs) and wait for TICK interrupt. A short pre-start time ( a few clock cycles) is used when XOSC16M is not enabled as controlled by CLKCTRL bit 5 and 4 (please refer to <b>Table 58 on page 104</b>). See also footnote for bit 7 regarding wakeup in <b>Table 58 on page 104</b>.</li> </ul> <p>Start-up time: As for Register retention, timers off. If awoken from TICK, a short pre-start time is used when XOSC16M is not enabled as controlled by CLKCTRL bit 5 and 4 (refer to <b>Table 58 on page 104</b>) or when XOSC16M is on in the Register retention mode (CLKCTRL bit 7). The short pre-start time will not be used if entering power-down before XOSC16M is running (this can be observed by polling bit 3 in CLKLFCTRL).</p> <p>Comment: Wakeup does not lead to system reset (after wakeup, program execution will resume from the current instruction).</p>
Standby	<p>Current: See <b>Table 85 on page 141</b></p> <p>Powered functions: In addition to Register retention:</p> <ul style="list-style-type: none"> <li>Program memory and Data memory</li> <li>VREG</li> <li>XOSC16M</li> </ul> <p>Wakeup source(s): In addition to Register retention:</p> <ul style="list-style-type: none"> <li>The interrupt sources RFIRQ and MISCIRQ (see <b>section 9.3 on page 93</b> and <b>11.3.5 on page 107</b>).</li> </ul> <p>Start-up time: ~ 100 ns</p> <p>Comment: Processor in standby, that is, clock stopped. I/O functions may be active.</p>

Mode	Brief description
Active	<p>Current: See <b>Table 85 on page 141</b></p> <p>Powered functions: Everything powered</p> <p>Wakeup source(s): -</p> <p>Start-up time: -</p> <p>Comment: Processor active and running</p>

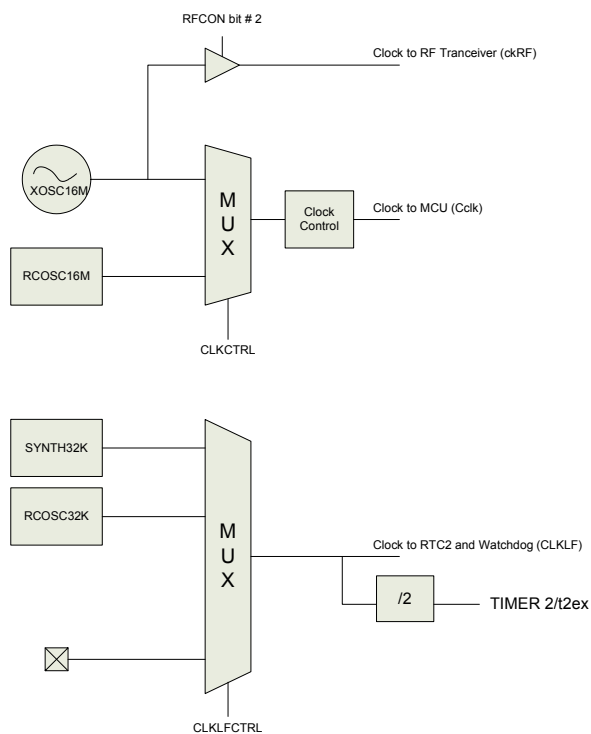
1. Please note that both Register retention power-down modes are entered by writing '100' to the PWRDWN register (refer to **Table 60 on page 106**). "Register retention timers on" is obtained by choosing an active CLKLF source as controlled by CLKLFCTRL[2:0] (refer to **Table 59 on page 105**).

**Table 57** Modes of operation

## 11.3 Functional description

### 11.3.1 Clock control

The clock to the MCU (Cclk) is sourced from either an on-chip RC oscillator or a crystal oscillator (see *chapter 13 on page 112*) for details.



**Figure 45** The clock system

The source and frequency of the clock to the microcontroller system is controlled by the CLKCTRL register:

Addr	Bit	R/W	Function	Reset value: 0x00
0xA3	7 <sup>1</sup>	R/W	1: Keep XOSC16M on in Register retention mode	
	6	R/W	1: Clock sourced directly from pin (XC1), bypass oscillators. <sup>2</sup> 0: Clock sourced by XOSC16M or RCOSC16M, see bit 3	
	5:4 <sup>3</sup>	R/W	00: Start both XOSC16M and RCOSC16M. <sup>4</sup> 01: Start RCOSC16M only. 10: Start XOSC16M only. 11: Reserved	
	3	R/W	1: Enable wakeup and interrupt (X16IRQ) from XOSC16M active 0: Disable wakeup and interrupt from XOSC16M active	
	2:0	R/W	Clock frequency to microcontroller system: 000: 16 MHz 001: 8 MHz 010: 4 MHz 011: 2 MHz 100: 1 MHz 101: 500 kHz 110: 250 kHz 111: 125 kHz	

1. If this bit is set to '1', it is necessary to write '00' to bit 5:4 before entering Register retention mode. Otherwise, wakeup may not function properly.
2. For test and application development involving no radio link, only. Must be cleared in normal operation.
3. When writing to this bit, the device must be put in Register retention for effects to occur.
4. Default setting, both oscillators started. Clock sourced from RCOSC16M initially and automatically switched to XOSC16M.

**Table 58** CLKCTRL register



The source of the 32 kHz clock (CLKLF) is controlled by the CLKLFCTRL register:

Addr	Bit	R/W	Function	Reset value: 0x07
0xAD	7	R	1: Read CLKLF (phase).	
	6	R	1: CLKLF ready to be used	
	5	-	Reserved	
	4	-	Reserved	
	3	R	1: Clock sourced by XOSC16M (that is, XOSC16M active/running) 0: Clock sourced by RCOSC16M	
	2:0	R/W	Source for CLKLF: 000: Reserved 001: RCOSC32K 010: Synthesized from XOSC16M when active, off otherwise 011: Reserved 100: From I/O pin (digital rail-to-rail signal) 101: Reserved 110: Reserved 111: None selected	

**Table 59** CLKLFCTRL register

**Note:** If a source for CLKLF is selected, the MCU system will not start unless CLKLF is operative. For example, when selecting CLKLF from I/O pin the external clock must be active for the MCU to wake up by pin.

### 11.3.2 Power down control – PWRDWN

The PWRDWN register is used by the MCU to set the system to a power saving mode:

Addr	Bit	R/W	Function	Reset value: 0x00
0xA4	7	R	Indicates a wakeup from pin if set This bit is either cleared by a read or by entering a power down mode	
	6	R	Indicates a wakeup from TICK if set This bit is either cleared by a read or by entering a power down mode	
	5	-	Reserved	
	4:3		Reserved	
	2:0	W	Set system to power down if different from 000 001: reserved 010: reserved 011: reserved 100: set system to Register retention 101: reserved 110: reserved 111: set system to standby (stop MCU clock)	
		R	Shows previous power down mode 000: Power off 001: reserved 010: reserved 011: reserved 100: Register retention 101: reserved 110: reserved 111: standby	

**Note:** On wakeup from powerdown, the PWRDWN register, bits 2:0, should be reset to 0x00. The content of the PWRDWN register can be read out first if needed.

*Table 60 PWRDWN register*

### 11.3.3 Operational mode control - OPMCON

The OPMCON register is used to control special behavior in some of the operation modes:

Addr	Bit	R/W	Function	Reset value: 0x00
0xAE	7:3	-	Reserved (always write '0' to these bits)	
	2	R/W	1: Subset of wakeup pins have active high polarity 0: All wakeup pins have active low polarity. Refer to section 11.3.6 on page 108.	
	1	R/W	Retention latch control 0: Latch open – pass through 1: Latch locked	
	0	R/W	Watchdog reset enable 0: If the on-chip watchdog functionality is enabled it will keep running. 1: The on-chip watchdog functionality will enter its reset state (wd off) when the operational mode Register Retention is entered.	

*Table 61 OPMCON register*

**Note:** If the Watchdog reset enable bit is enabled, you must wait at least until the first negative edge of CLKLF after enabling CLKLF before proceeding to Register Retention. Waiting for the negative edge of CLKLF is not needed when entering into any other power-down state or if the Watchdog reset enable bit is not enabled.

### 11.3.4 Reset result – RSTREAS

There are four reset sources that initiate the same reset/ start-up sequence. These are:

- Reset from the on chip reset generator
- Reset from pin
- Reset generated from the on chip watchdog function

The RSTREAS register stores the reason for the last reset, all cleared indicates that the last reset was from the on-chip reset generator. A write operation to the register will clear all bits. Unless cleared after read (by on-chip reset or by a write operation), RSTREAS will be cumulative.

Addr	Bit	R/W	Function
0xB1	7:3	-	Not used
	2:0	R	000: On-chip reset generator 001: RST pin 010: Watchdog

*Table 62 RSTREAS register*

### 11.3.5 Wakeup configuration register – WUCON

The following wakeup sources is available in STANDBY power down mode.

Addr	Bit	R/W	Function	Reset value 0x00
0xA5	7:6	RW	00: Enable wakeup on RFIRQ if interrupt is enabled (IEN1.1=1) 01: Reserved, not used 10: Enable wakeup on RFIRQ 11: Ignore RFIRQ	
	5:4	RW	00: Enable wakeup on TICK (from RTC2) if interrupt is enabled (IEN1.5=1) 01: Reserved, not used 10: Enable wakeup on TICK 11: Ignore TICK	
	3:2	RW	00: Enable wakeup on WUOPIRQ if interrupt is enabled (IEN1.3=1) 01: Reserved, not used 10: Enable wakeup on WUOPIRQ 11: Ignore WUOPIRQ	
	1:0	RW	00: Enable wakeup on MISCIRQ if interrupt is enabled (IEN1.4=1) 01: Reserved, not used 10: Enable wakeup on MISCIRQ 11: Ignore MISCIRQ	

*Table 63 WUCON register*

MISCIRQ is set if one of the following take place:

- XOSC16M has started and is ready to be used.
- ADC finished with conversion, and data ready.

### 11.3.6 Pin wakeup configuration

Pin wakeup is configured by two registers, WUOPC1 and WUOPC2

Address (Hex)	Name/Mnemonic	Bit	Reset value	Type	Description
0xCE	WUOPC1	7:0	0x00	R/W	Wake Up On Pin configuration register 1. n = 1: Wake up on pin enabled. n = 0: Wake up on the corresponding pin disabled.
0xCF	WUOPC0	7:0	0x00	R/W	Wake Up On Pin configuration register 0. n = 1: Wake up on pin enabled. n = 0: Wake up on the corresponding pin disabled.

**Table 64** WUOPCx registers

The following table shows which port-pin/ gpio that give wakeup if the corresponding enable bit in the WUOPCx register is asserted. Pins marked with an asterisk have selectable polarity controlled by OPMCON[2]. All other pins have low polarity.

WUOPC bit	nRF31512 wakeup pins
WUOPC1[1]	P1 . 1
WUOPC1[0]	P1 . 0
WUOPC0[7]	P0 . 7
WUOPC0[6]	P0 . 6*
WUOPC0[5]	P0 . 5
WUOPC0[4]	P0 . 4
WUOPC0[3]	P0 . 3
WUOPC0[2]	P0 . 2
WUOPC0[1]	P0 . 1
WUOPC0[0]	P0 . 0

**Table 65** Configuration of pin wakeup

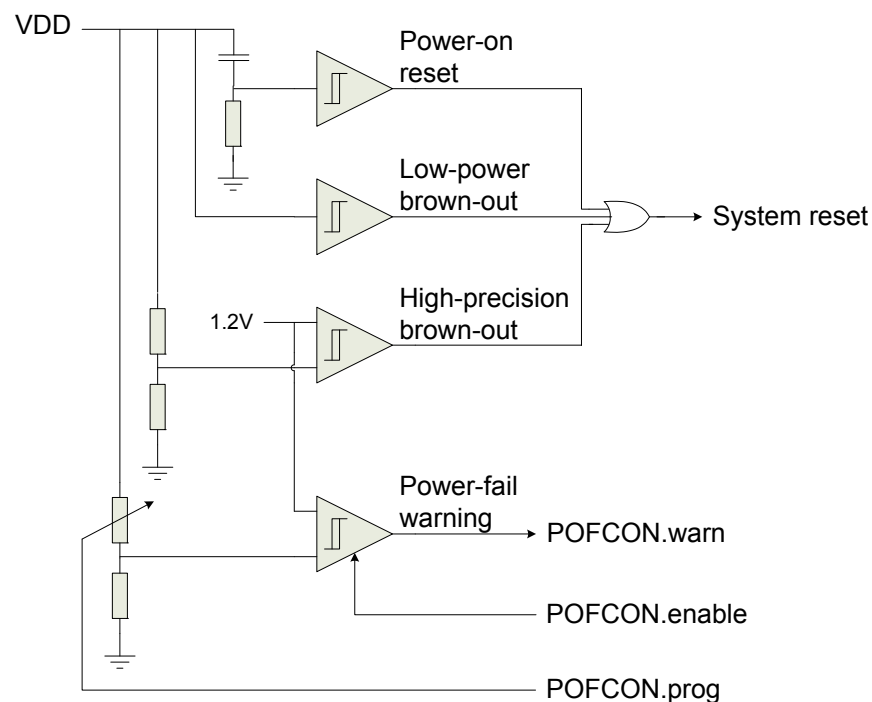
## 12 Power supply supervisor

The power supply supervisor initializes the system at power-on, provides an early warning of impending power failure, and puts the system in reset state if the supply voltage is too low for safe operation.

### 12.1 Features

- Power-on reset with timeout delay
- Brown-out reset
- Power-fail warning with programmable threshold, interrupt and hardware protection of data in program memory

### 12.2 Block diagram

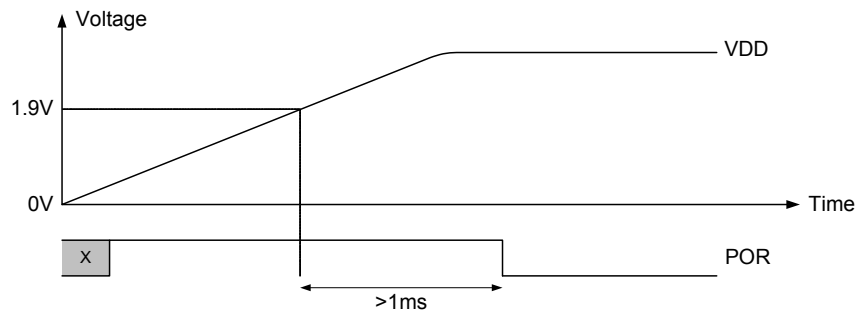


**Figure 46** Block diagram of power supply supervisor

### 12.3 Functional description

#### 12.3.1 Power-on reset

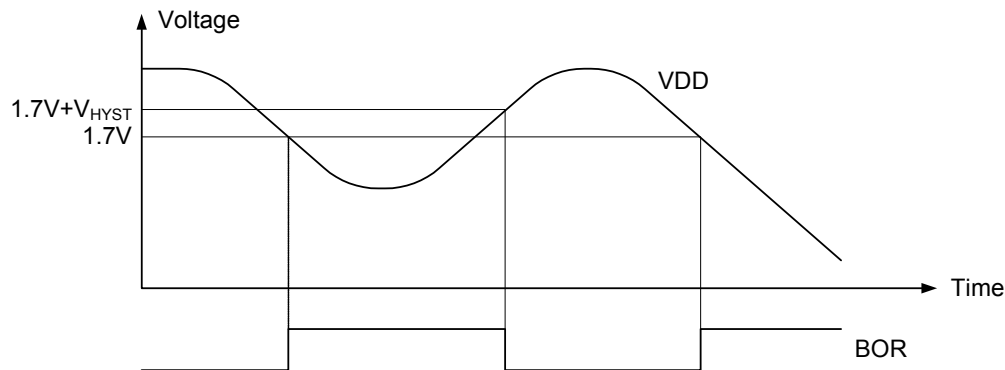
The Power-On Reset (POR) generator initializes the system at power-on. It is based on an RC network and a comparator, as illustrated in **Figure 46**. For proper operation the supply voltage should rise monotonically with rise time according to the specifications in **Table 83 on page 137**. The system is held in reset state for at least 1ms after the supply has reached the minimum operating voltage of 1.9V.



**Figure 47** Power-on reset

### 12.3.2 Brown-out reset

The Brown-Out Reset (BOR) generator puts the system in reset state if the supply voltage drops below the BOR threshold, which is about 1.7 V. There is approximately 70 mV of hysteresis ( $V_{\text{HYST}}$ ). This means that if a reset is triggered when the supply voltage drops below 1.7 V, the supply must rise above 1.77 V again before the device becomes operational. Hysteresis prevents the comparator output from oscillating when VDD is close to threshold. The BOR generator is enabled when the system is in active or standby mode.



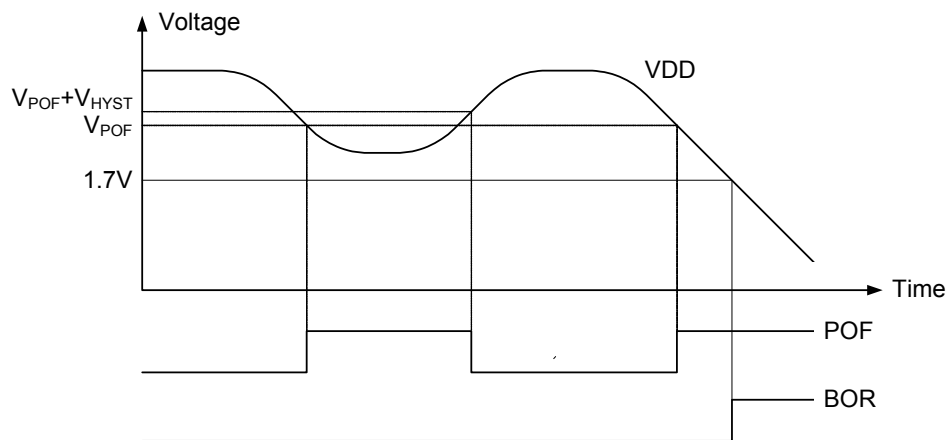
**Figure 48** Brown-Out Reset

### 12.3.3 Power-fail comparator

The Power-Fail (POF) comparator provides the MCU with an early warning of impending power failure. It will not reset the system, but gives the MCU time to prepare for an orderly power-down. It also provides hardware protection of data stored in program memory, by preventing write instructions from being executed. Refer to [section 6.3.3 on page 72](#) for details.

The POF comparator is enabled or disabled by writing the **enable** bit in the POFCON register (see [Table 66 on page 111](#)). When enabled, it will be powered up when the system is in active or standby mode. The **warn** bit is set to '1' if the supply voltage is below the programmable threshold. An interrupt (POFIRQ) is also produced. Write instructions to program memory will not be executed as long as **warn** is '1'.

Use the **prog** bits to configure the desired threshold voltage ( $V_{\text{POF}}$ ). The available levels are 2.0, 2.1, 2.2 and 2.3 V, defined for falling supply voltage. The comparator has approximately 0.05 V of hysteresis ( $V_{\text{HYST}}$ ).



**Figure 49** Power-fail comparator

## 12.4 SFR registers

Addr	Bit	Name	RW	Function	Reset value: 0x00
0xDC	7	enable	RW	POF enable: 0: Disable POF comparator 1: Enable POF comparator	
	6:5	prog	RW	POF threshold: 00: 2.0 V 01: 2.1 V 10: 2.2 V 11: 2.3 V	
	4	warn	R	POF warning: 0: VDD above threshold 1: VDD below threshold	
	3:0	-	-	Not used	

**Table 66** POFCON register

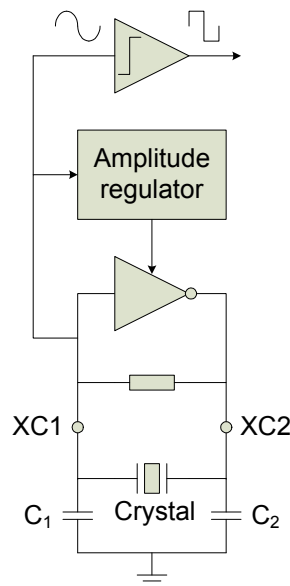
## 13 On-chip oscillators

There are two high frequency oscillators and one low frequency oscillator. The primary high frequency clock source is a 16 MHz crystal oscillator. There is also a fast starting 16 MHz RC oscillator, which is used primarily to provide the system with a high frequency clock while it is waiting for the crystal oscillator to start up. The low frequency clock is supplied by a 32.768 kHz RC oscillator. External 16 MHz and 32.768 kHz clocks may also be used instead of the on-chip oscillators. See [section 11.3.1 on page 103](#) for control of the clock sources.

### 13.1 Features

- Low-power amplitude regulated 16 MHz crystal oscillator
- Fast starting 16 MHz RC oscillator with  $\pm 5\%$  frequency accuracy
- Ultra low-power 32.768 kHz RC oscillator with  $\pm 10\%$  frequency accuracy

### 13.2 Block diagrams



**Figure 50** Block diagram of 16 MHz crystal oscillator

## 13.3 Functional description

### 13.3.1 16 MHz crystal oscillator

The 16 MHz crystal oscillator (XOSC16M) is designed to be used with an AT-cut quartz crystal in parallel resonant mode. To achieve correct oscillation frequency it is very important that the load capacitance matches the specification in the crystal datasheet. The load capacitance,  $C_L$ , as specified in the crystal datasheet, is the total capacitance seen by the crystal across its terminals:



$$C_{LOAD} = \frac{C'_1 \cdot C'_2}{C'_1 + C'_2}$$

$$C'_1 = C_1 + C_{PCB1} + C_{PIN}$$

$$C'_2 = C_2 + C_{PCB2} + C_{PIN}$$

$C_1$  and  $C_2$  are ceramic SMD capacitors connected between each crystal terminal and VSS,  $C_{PCB1}$  and  $C_{PCB2}$  are stray capacitances on the PCB, while  $C_{PIN}$  is the input capacitance on the **xc1** and **xc2** pins (typically 4 pF).  $C_1$  and  $C_2$  should be of the same value, or as close as possible.

To ensure a functional radio link the frequency accuracy must be  $\pm 60$  ppm or better. The initial tolerance of the crystal, drift over temperature, aging and frequency pulling due to incorrect load capacitance must all be taken into account. For reliable operation the crystal load capacitance, shunt capacitance, equivalent series resistance (ESR) and drive level must comply with the specifications in **Table 85 on page 141**. It is recommended to use a crystal with lower than maximum ESR if the load capacitance and/or shunt capacitance is high. This will give faster start-up and lower current consumption.

The start-up time is typically about 1 ms for a crystal with 9 pF load capacitance and an ESR specification of 60  $\Omega$  max. This value is valid for crystals in a 3.2x2.5 mm can. If you use the smaller crystal cans (like 2.0x2.5 mm), pay particular attention to the startup time of the crystal. These crystals have a longer startup than crystals in larger cans. To make sure the startup time is <1.5 ms use a crystal for load capacitance of 6pF. A low load capacitance will reduce both startup time and current consumption. For more details regarding how to measure the startup of a specific crystal, please see the nAN24-13 application note.

The crystal oscillator is normally running only when the system is in active or standby mode. It is possible to keep it on in register retention mode as well, by writing a '1' to bit 7 in the CLKCTRL register (see **Table 58 on page 104**). This is recommended if the system is expected to wake up again in less than 5 ms. The reason is that the additional current drawn during start-up makes it more power-efficient to let the oscillator run for a few extra milliseconds than to restart it.

### 13.3.2 16 MHz RC oscillator

The 16 MHz RC oscillator (RCOSC16M) is used primarily to provide a high speed clock while the crystal oscillator is starting up. It starts in just a few microseconds, and has a frequency accuracy of  $\pm 5\%$ .

By default, the 16 MHz RC and crystal oscillators are started simultaneously. The RC oscillator supplies the clock until the crystal oscillator has stabilized. The system then makes an automatic switch to the crystal oscillator clock, and turns off the RC oscillator to save power. Bit 3 in the CLKCTRL register can be polled to check which oscillator is currently supplying the high speed clock.

The system can be configured to start only one of the two 16 MHz oscillators. Write bit 4 and 5 in the CLKCTRL register to choose the desired behavior. Note that the RF transceiver cannot be used while the high frequency clock is sourced by the RC oscillator. The ADC may also have reduced performance.

### 13.3.3 External 16 MHz clock

The device may be used with an external 16MHz clock applied to the **xc1** pin. Write a '1' to bit 6 in the CLKCTRL register if the external clock is a rail-to-rail digital signal. (This is for test and application development involving no radio link, only. Bit 6 must be cleared in normal operation.) The input signal may also be analog, coming from e.g. the crystal oscillator of a microcontroller. In this case the crystal oscillator must also be enabled, since it is used to convert the analog input into a digital clock signal. CLKCTRL[6] must be '0', and CLKCTRL[5:4] must be '10' to enable the oscillator. An input amplitude of 0.8V peak-to-peak

or higher is recommended to achieve low current consumption and a good signal-to-noise ratio. The DC level is not important as long as the applied signal never rises above VDD or drops below VSS. The **XC1** pin will load the microcontrollers crystal with approximately 2.5 pF in addition to PCB routing. **XC2** shall not be connected.

**Note:** A frequency accuracy of  $\pm 60$  ppm or better is required to get a functional radio link.

### 13.3.4 32.768 kHz RC oscillator

The low frequency clock may be generated by a 32.768 kHz RC oscillator (RCOSC32K). The 32.768 kHz RC oscillator is enabled by writing '001' to CLKLFCTRL[2:0]. It typically starts in less than 0.5 ms. Bit 6 in the CLKLFCTRL register can be polled to check if the oscillator is ready for use.

### 13.3.5 Synthesized 32.768 kHz clock

The low frequency clock can also be synthesized from the 16 MHz crystal oscillator clock. Write '010' to CLKCTRL[2:0] to select this option. The synthesized clock will only be available in system modes where the 16 MHz crystal oscillator is active. (This will be possible in the operational modes "Register retention," "Standby," and "Active.")

### 13.3.6 External 32.768 kHz clock

The device may be used with an external rail-to-rail digital 32.768 kHz clock applied to the **P0.1** port pin. Write '100' to CLKCTRL[2:0] to select this option.

## 14 Encryption/decryption accelerator

You can utilize the on-chip encryption/decryption accelerator for more time and power effective firmware. The accelerator is an 8 by 8 Galois Field Multiplier with an 8 bits output. The following polynomial is used:

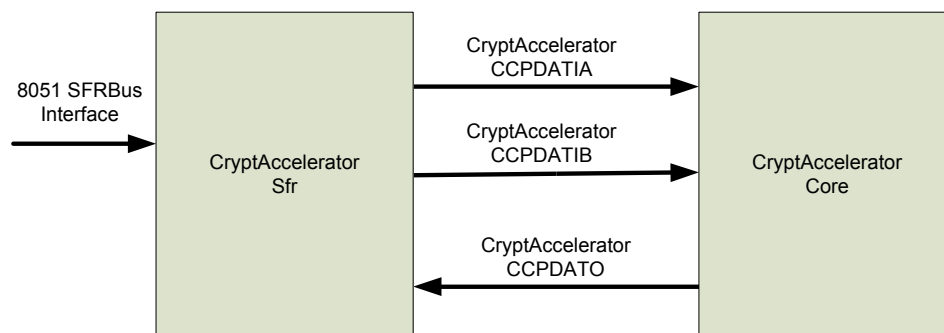
$$m(x) = x^8 + x^4 + x^3 + x + 1$$

This is the polynomial used by AES (Advanced Encryption Standard).

### 14.1 Features

- AES encryption
- 8 bit output
- Firmware for 128 bit AES available from Nordic Semiconductor.
- The result from the co-processing is available one clock period after the input data registers have changed.

### 14.2 Block diagram



**Figure 51** Encryption/decryption accelerator

### 14.3 Functional description

The following registers control the encryption/decryption accelerator.

Address (Hex)	Name/Mnemonic	Bit	Reset values	Type	Description
0xDD	CCPDATIA	7:0	0x00	R/W	Encryption/decryption accelerator data in register A.
0xDE	CCPDATIB	7:0	0x00	R/W	Encryption/decryption accelerator data in register B.
0xDF	CCPDATO	7:0	0x00	R	Encryption/decryption accelerator data out register.

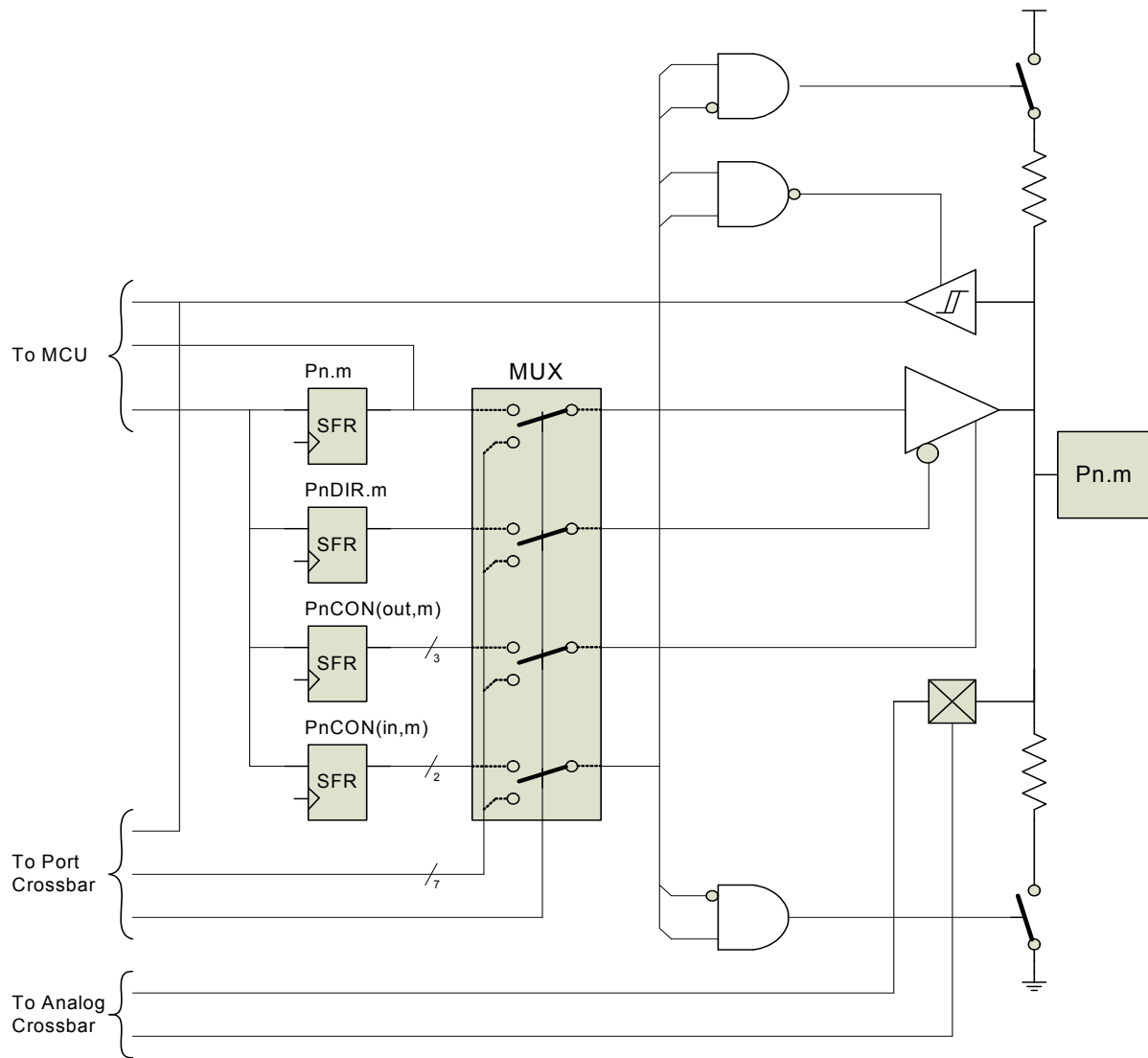
**Table 67** Encryption/decryption accelerator registers

The two registers CCPDATIA and CCPDATIB contain the input data, whilst CCPDATO contains the result from the co-processing. CCPDATO is updated one clock period after one of the input data registers has changed.

## 15 General purpose I/O port and pin assignments

The I/O pins are default set to general purpose I/O for the MCU. The numbers of available I/Os is 10 for the nRF31512 32 pin 5x5 mm. The I/O pins are also shared with I/O requirements from peripheral blocks like SPI as well as more specialized functions. Connections between these other peripheral blocks and the pins are made dynamically by the PortCrossbar module.

### 15.1 Block diagram



**Figure 52** I/O pin circuitry block diagram

## 15.2 Functional description

### 15.2.1 General purpose I/O pin functionality

Each of the I/O pins has a generic control functionality that sets pin features for the GPIO of the MCU.

The features offered by the pins include:

- Digital or Analog
- Configurable Direction
- Configurable Drive Strength
- Configurable Pull Up/Down

This functionality is multiplexed with the functionality of the PortCrossbar module which takes control and configures the pins depending on the needs of the peripheral block connected. The pin circuitry is shown in **Figure 52 on page 117**.

The pins are connected by default to a pin Multiplexer (MUX) that is connected to the GPIO registers of the MCU. Register  $Pn.m$  (n-port number, m - bit number) contains MCU GPIO data,  $PDIRn.m$  register controls input/output direction and  $PCONn.m$  register controls pin features drive strength and pull up/down resistors for each pin.

When the MCU enables one of the peripheral blocks the pin MUX disconnects the MCU control of the pin and hands control over to the PortCrossbar module to set direction and pin features.

However, if the pin is operated as an analog input, the MCU must set the pin control registers  $PDIR$  and  $PCON$  separately to prevent conflicts between pin configuration and the needs of the analog peripheral blocks.

There is one set of  $Pn.m$ ,  $PnDIRm$  and  $PnCONn$  for each port.  $Pn.m$  and  $PnDIRm$  control only one parameter each, this means that a write/read operation to them controls/reads the status of the port directly. However, to control or read the features of a pin you use the  $PnCONm$  to write/read to one pin at a time. The  $PnCON$  register contains an address for the pin, information on whether it is an input or an output feature that is to be updated and the feature that is to be enabled.

The features available:

- Output buffer on, normal drive strength
- Output buffer on, high drive strength
- Input buffer on, no pull up/down resistor
- Input buffer on, pull up resistor
- Input buffer on, pull down resistor
- Input buffer off

Example: If four pins in port 0 are set as inputs with the pull up resistor enabled, then this is done with one write to  $P0DIR$  and four write operations to  $P0CON$  and only updating the pin address in  $P0CON$  for each write.

## 15.2.2 PortCrossbar functionality

The PortCrossbar sets up connections between the I/O pins and the peripheral block of the device.

### 15.2.2.1 Dynamic allocation of pins

The PortCrossbar modifies connections dynamically based on run-time variations in system needs of the peripheral blocks (SPI, 2 wire etc) of the device. This feature is necessary because the number of available pins is small compared to the combined I/O needs of all the peripheral blocks. Consequently, on the smaller package options there may be conflicting pin assignments. These are resolved through a set of priorities assigned to each peripheral block. The pin out tables for each package option can be seen in *Table 68 on page 120*, *Table 68 on page 121* and *Table 70 on page 126*.

#### 15.2.2.2 Dynamic pin allocation for digital blocks

Each digital peripheral block that needs an I/O is represented in the pin out tables with the interface names of the block and the direction enforced on each pin. The priority of the blocks relative to potentially conflicting blocks is also shown. If the block is enabled, and no higher priority block is enabled, all the I/O needs are granted. The PortCrossbar never grants partial fulfilment of a digital I/O request even if a conflict exists only for some of the pins. A requesting digital device gets all or none of its I/O needs granted.

#### 15.2.2.3 Dynamic pin allocation for analog blocks

A dynamic request for analog I/O is similar to that of a digital I/O. However, for analog blocks only the interface signals actually used as inputs to the analog blocks, configured by ADCCON1.chsel and ADCCON1.refsel, are connected to a device pin. This is different from the digital peripheral blocks where all the I/O of a block are reserved once the block is enabled.

**Note:** The implementation does not prevent simultaneous digital and analog use of a pin. If a pin is to be used for analog input, digital I/O buffers and digital peripheral blocks connected to the same pin should normally be disabled. Conflicts between analog blocks are resolved through priority.

If analog functionality is enabled for a pin, this is done without modifying or disabling the pins digital configuration. If particular digital input and/or output configuration are necessary for an analog pin to function correctly, this configuration must be enabled in registers PxCON and PxDIR separately, before enabling the analog block.

#### 15.2.2.4 Default pin allocation

If no peripheral blocks request I/O, a default pinout as listed in the default column in the pin out maps are enabled. This means that all device pins are used for MCU GPIO. After reset, all I/Os are configured to be digital inputs. The features, direction and I/O data on the pins are in this case controlled by registers PnCON, PnDIR and Pn.

## 15.3 I/O pin maps

The following conventions are used in all pin out maps:

- For dynamic connections of digital peripheral blocks, the direction of each pin is indicated by 'in', 'out' or 'inout' next to the interface name.
- Dynamic analog connections are indicated with 'ana'.
- Digital peripheral blocks with potentially conflicting I/O needs are highlighted with blue background in the pinout tables.
- For blocks marked with a green background, conflicts may exist with other green and blue devices, depending on the configuration. Please refer to the documentation of the configurable (green) blocks for information on how the configuration affects the I/O usage.



### 15.3.1 Pin assignments for nRF31512 in 32 pin package

Pins **P0.5** to **P1.0** have two system inputs listed per pin. This means that the input from the pin is driving both block inputs if the pin is configured as an input.

The **FMISO** pin driver is enabled only when **FCSN** is active.

Pin	Default connections		Dynamically enabled connections			
	Inputs	Outputs	CLKLF	SPI Master	OTP Slave SPI	ADC
			priority 1	priority 2	priority 3	priority 4
<b>P1.1</b>	<b>p1Di.1</b>	<b>p1Do.1</b>				
<b>P1.0</b>	<b>p1Di.0</b> ..... T1	<b>p1Do.0</b>				
<b>P0.7</b>	<b>p0Di.7</b> ..... T0	<b>p0Do.7</b>			<b>FCSN</b> <sup>1</sup>	in
<b>P0.6</b>	<b>p0Di.6</b> ..... GPINT1	<b>p0Do.6</b>			<b>FMISO</b> <sup>1</sup>	out
<b>P0.5</b>	<b>p0Di.5</b> ..... GPINT0	<b>p0Do.5</b>			<b>FMOSI</b> <sup>1</sup>	in
<b>P0.4</b>	<b>p0Di.4</b>	<b>p0Do.4</b>			<b>FCK</b> <sup>1</sup>	in
<b>P0.3</b>	<b>p0Di.3</b>	<b>p0Do.3</b>		<b>MMISO</b>	in	
<b>P0.2</b>	<b>p0Di.2</b>	<b>p0Do.2</b>		<b>MMOSI</b>	out	
<b>P0.1</b>	<b>p0Di.1</b>	<b>p0Do.1</b>	<b>CLKLF</b> <sup>2</sup>	<b>MSCK</b>	out	
<b>P0.0</b>	<b>p0Di.0</b>	<b>p0Do.0</b>				
Conflict may exist depending on device configuration. In the case of a conflict, use priorities to determine I/O allocation						

1. OTP SPI interface only activated when **PROG** is set higher than VIH for the **PROG** pin, no conflict with runtime operations.
2. Connection depends on configuration register **CLKLFCTRL[2:0]**.  
**CLKLFCTRL[2:0] = 100**: Digital clock source for **CLKLF**.

**Table 68** Pin out map for the nRF31512 in 32 pin package

### 15.3.2 Programmable registers

Desired pin direction and functionality is configured using the configuration registers **P0DIR**, **P1DIR**, collectively referred to as **PxDIR**, and **P0CON**, **P1CON**, referred to as **PxCON**. The **PxDIR** registers determine the direction of the pins and the **PxCON** registers contain the functional options for input and output pin operation.

The PortCrossbar by default (at reset) configures all pins as inputs and connects them to the MCU GPIO (pxDi).

To change pin direction, write the desired direction to the PxDIR registers.

Register name: P0DIR			Address: 0x93	Reset value: 0xFF
Bit	Name	RW	Function	
7:0	dir	RW	Direction bits for pins <b>P0 . 0</b> – <b>P0 . 7</b> . Output: dir = 0, Input: dir = 1.  P0DIR 0 - <b>P0 . 0</b> P0DIR 1 - <b>P0 . 1</b> P0DIR 2 - <b>P0 . 2</b> P0DIR 3 - <b>P0 . 3</b> P0DIR 4 - <b>P0 . 4</b> P0DIR 5 - <b>P0 . 5</b> P0DIR 6 - <b>P0 . 6</b> P0DIR 7 - <b>P0 . 7</b>	

*Table 69 P0DIR register*

Register name: P1DIR			Address: 0x94	Reset value: 0xFF
Bit	name	RW	Function	
7:0	dir	RW	Direction bits for pins <b>P1 . 0</b> – <b>P1 . 1</b> . Output: dir = 0, Input: dir = 1. P1DIR 0 - <b>P1 . 0</b> P1DIR 1 - <b>P1 . 1</b>	

*Table 70 P1DIR register*

The input and output options of each pin are configured in the PxCON registers. The PxCON registers have to be written once per pin (one write operation to the PxCON register configures the input/output options of a selected pin in the port).

To read the current input or output options for a pin, you first need to perform a write operation to retrieve the desired bit address and option type (input or output).

For instance, to read the output mode of pin **P0.5**: Write to P0CON with a bitAddr value of 101, a readAddr value of 1 and a inOut value of 0 (output). Then read from P0CON. The output mode of pin 5 is now found in bits 7:5 of the read data.

Register name: P0CON			Address: 0x9E	Reset value: 0x00
Bit	Name	RW	Function	
7:5	pinMode	RW	<p>Functional input or output mode for pins <b>P0.0</b> – <b>P0.7</b>.</p> <p>For a write operation: The functional mode you would like to write to the pin. The inOut field determines if the input or output mode is written, the bitAddr field determines which pin is affected.</p> <p>Output modes using bits 7:5:</p> <ul style="list-style-type: none"> <li>000 Digital output buffer normal drive strength</li> <li>011 Digital output buffer high drive strength</li> <li>(all other value combinations are illegal)</li> </ul> <p>Input modes using bits 6:5:</p> <ul style="list-style-type: none"> <li>00 Digital input buffer on, no pull up/down resistors</li> <li>01 Digital input buffer on, pull down resistor connected</li> <li>10 Digital input buffer on, pull up resistor connected</li> <li>11 Digital input buffer off</li> </ul> <p>For a read operation: The current functional mode of the pin. The inOut field determines if the input or output mode is reported, while the bitAddr field indicates which pin is selected.</p>	
4	inOut	W	<p>This bit indicates if the current write operation relates to the input or output configuration of the addressed pin.</p> <ul style="list-style-type: none"> <li>inOut = 0 - Operate on the output configuration</li> <li>inOut = 1 - Operate on the input configuration</li> </ul>	
3	readAddr	W	<p>If this bit is set, the purpose of the current write operation is to provide the bit address for later read operations. Consequently, the value of the bitAddr field is saved. The value of the inOut field is also saved, determining if the input or output mode is to be read. The pinMode field is ignored when readAddr is set.</p> <p>If this bit is not set, the pin mode of the addressed pin is updated with the value of the pinMode field. The inOut field determines if the input or output mode is updated.</p> <p><b>Note:</b> There is only one pair of inOut and bitAddr values that is shared with P1CON registers. P0CON should be read without any intervening write access to P0CON, P1CON with readAddr set.</p>	
2:0	bitAddr	W	<p>If the readAddr bit is set, the value of the bitAddr field is stored. For subsequent read operations from P0CON, the pin for which the pinMode will be returned is given by the list below.</p> <ul style="list-style-type: none"> <li>bitAddr = 000 - <b>P0.0</b></li> <li>bitAddr = 001 - <b>P0.1</b></li> <li>bitAddr = 010 - <b>P0.2</b></li> <li>bitAddr = 011 - <b>P0.3</b></li> <li>bitAddr = 100 - <b>P0.4</b></li> <li>bitAddr = 101 - <b>P0.5</b></li> <li>bitAddr = 110 - <b>P0.6</b></li> <li>bitAddr = 111 - <b>P0.7</b></li> </ul>	

**Table 71** P0CON register

Register name: P1CON			Address: 0x9F	Reset value: 0x00
Bit	Name	RW	Function	
7:5	pinMode	RW	<p>Functional input or output mode for pins <b>P1.0</b> – <b>P1.1</b>.</p> <p>For a write operation: The functional mode you would like to write to the pin. The inOut field determines if the input or output mode is written, the bitAddr field determines which pin is affected.</p> <p>Output modes using bits 7:5:</p> <ul style="list-style-type: none"> <li>000 Digital output buffer normal drive strength</li> <li>011 Digital output buffer high drive strength</li> <li>(all other value combinations are illegal)</li> </ul> <p>Input modes using bits 6:5:</p> <ul style="list-style-type: none"> <li>00 Digital input buffer on, no pull up/down resistors</li> <li>01 Digital input buffer on, pull down resistor connected</li> <li>10 Digital input buffer on, pull up resistor connected</li> <li>11 Digital input buffer off</li> </ul> <p>For a read operation: The current functional mode of the pin. The inOut field determines if the input or output mode is reported, while the bitAddr field indicates which pin is selected.</p>	
4	inOut	W	<p>This bit indicates if the current write operation relates to the input or output configuration of the addressed pin.</p> <ul style="list-style-type: none"> <li>inOut = 0 - Operate on the output configuration</li> <li>inOut = 1 - Operate on the input configuration</li> </ul>	
3	readAddr	W	<p>If this bit is set, the purpose of the current write operation is to provide the bit address for later read operations. Consequently, the value of the bitAddr field is saved. The value of the inOut field is also saved, determining if the input or output mode is to be read. The pinMode field is ignored when readAddr is set.</p> <p>If this bit is not set, the pin mode of the addressed pin is updated with the value of the pinMode field. The inOut field determines if the input or output mode is updated.</p> <p><b>Note:</b> There is only one pair of inOut and bitAddr values that is shared with P0CON registers. P0CON should be read without any intervening write access to P0CON, P1CON with readAddr set.</p>	
2:0	bitAddr	W	<p>If the readAddr bit is set, the value of the bitAddr field is stored. For subsequent read operations from P1CON, the pin for which the pinMode will be returned, is given by the list below.</p> <ul style="list-style-type: none"> <li>bitAddr = 000 - <b>P1.0</b></li> <li>bitAddr = 001 - <b>P1.1</b></li> </ul>	

**Table 72** P1CON register

While the I/O ports are used as MCU GPIO, the pin values are read and controlled by the MCU port registers P1 to P0.

Address	Name	Bit	Reset value	Type	Description
0x90	P1	7:0	0xFF	R/W	Port 1 value
0x80	P0	7:0	0xFF	R/W	Port 0 value

*Table 73 P0-P1 registers*

## 16 SPI

nRF31512 features a double buffered Master Serial Peripheral Interface (SPI). You can configure it to work in all four SPI modes. The default is mode 0.

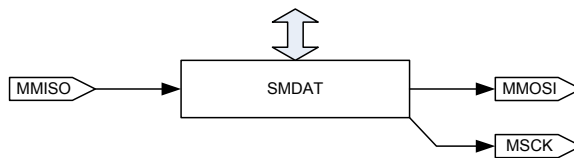
The SPI connects to the following pins of the device: **MMISO**, **MMOSI**, and **MSCK**.

The SPI Master function does not generate any chip select signal (CSN). The programmer typically uses another programmable digital I/O to act as chip selects for one or more external SPI Slave devices.

### 16.1 Features

- Double buffered FIFO
- Full-duplex operation
- Supports SPI modes 0 through 3
- Configurable data order on MMISO/MMOSI
- Four Master interrupt sources

### 16.2 Block diagram



**Figure 53** SPI Master

## 16.3 Functional description

### 16.3.1 SPI master

The following registers control the SPI master:

Address (Hex)	Name/mnemonic	Bit	Reset value	Type	Description
0xFC	SPIMCON0	6:0	0x20	R/W	SPI Master configuration register 0.
	<i>clockFrequency</i>	6:4	010	R/W	Frequency on MSCK. ckMCU is the MCU clock frequency.) 000: 1/2 ·ckMCU 001: 1/4·ckMCU 010: 1/8 ·ckMCU 011: 1/16·ckMCU 100: 1/32·ckMCU 101: 1/64·ckMCU 110: 1/64·ckMCU 111: 1/64·ckMCU
	<i>dataOrder</i>	3	0	R/W	Data order (bit wise per byte) on serial output and input (MMOSI and MMISO respectively). 1: LSBit first, MSBit last. 0: MSBit first, LSBit last.
	<i>clockPolarity</i>	2	0	R/W	Defines the SPI Master's operating mode together with SPIMCON0.1, see <b>section 16.3.2 on page 128</b> . 1: MSCK is active 'low'. 0: MSCK is active 'high'.
	<i>clockPhase</i>	1	0	R/W	Defines the SPI Master's operating mode together with SPIMCON0.2, see <b>section 16.3.1 on page 127</b> . 1: Sample on trailing edge of MSCK, shift on leading edge. 0: Sample on leading edge of MSCK, shift on trailing edge.
	<i>spiMasterEnable</i>	0	0	R/W	1: SPI Master is enabled. The clock to the SPI Master core functionality is running. An SPI transfer can be initiated by the MCU via the 8051 SFR Bus (TX). 0: SPI Master is disabled. The clock to the SPI Master core functionality stands still.
0xFD	SPIMCON1	3:0	0x0F	R/W	SPI Master configuration register 1.
	<i>maskIrqRxFifoFull</i>	3	1	R/W	1: Disable interrupt when RX FIFO is full. 0: Enable interrupt when RX FIFO is full.
	<i>maskIrqRxDataReady</i>	2	1	R/W	1: Disable interrupt when data is available in RX FIFO. 0: Enable interrupt when data is available in RX FIFO.
	<i>maskIrqTxFifoEmpty</i>	1	1	R/W	1: Disable interrupt when TX FIFO is empty. 0: Enable interrupt when TX FIFO is empty.
	<i>maskIrqTxFifoReady</i>	0	1	R/W	1: Disable interrupt when a location is available in TX FIFO. 0: Enable interrupt when a location is available in TX FIFO.
0xFE	SPIMSTAT	3:0	0x03	R	SPI Master status register.
	<i>rxFifoFull</i>	3	0	R	Interrupt source. 1: RX FIFO full. 0: RX FIFO can accept more data from SPI. Cleared when the cause is removed.

Address (Hex)	Name/mnemonic	Bit	Reset value	Type	Description
	<i>rxDataReady</i>	2	0	R	Interrupt source. 1: Data available in RX FIFO. 0: No data in RX FIFO. Cleared when the cause is removed.
	<i>txFifoEmpty</i>	1	1	R	Interrupt source. 1: TX FIFO empty. 0: Data in TX FIFO. Cleared when the cause is removed.
	<i>txFifoReady</i>	0	1	R	Interrupt source. 1: Location available in TX FIFO. 0: TX FIFO full. Cleared when the cause is removed.
0xFF	SPIMDAT	7:0	0x00	R/W	SPI Master data register. Accesses TX (write) and RX (read) FIFO buffers, both two bytes deep.

**Table 74** SPI Master registers

The SPI Master is configured through SPIMCON0 and SPIMCON1. It is enabled by setting SPIMCON0[0] to '1'. The SPI Master supports all four SPI modes, selected by SPIMCON0[2] and SPIMCON0[1] as described in **section 16.3.2 on page 128**. The bit wise data order per byte on MMISO/MMOSI is defined by SPIMCON0[3]. MSCK can run on one of six predefined frequencies in the range of 1/2 to 1/64 of the MCU clock frequency, as defined by SPIMCON0[6] down to SPIMCON0[4].

SPIMDAT accesses both the TX (write) and the RX (read) FIFOs, which are two bytes deep. The FIFOs are dynamic and can be refilled according to the state of the status flags: "FIFO ready" means that the FIFO can accept data. "Data ready" means that the FIFO can provide data, minimum one byte.

Four different sources can generate interrupt, unless they are masked by their respective bits in SPIMCON1. SPIMSTAT reveals which sources are active.

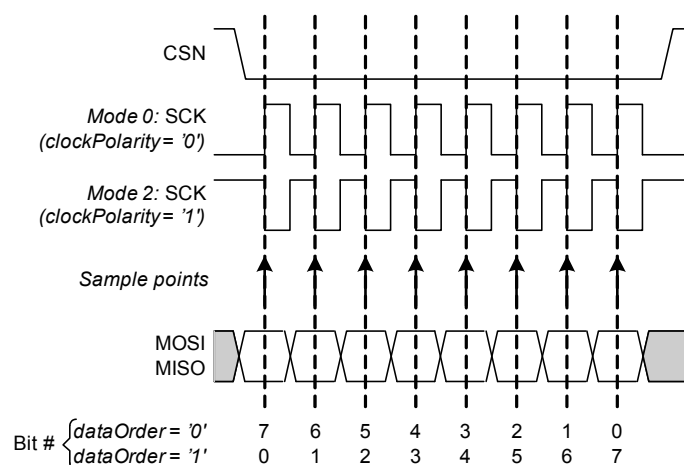
## 16.3.2 SPI timing

The four different SPI modes are presented in **Table 75**, **Figure 54** and **Figure 55 on page 129**

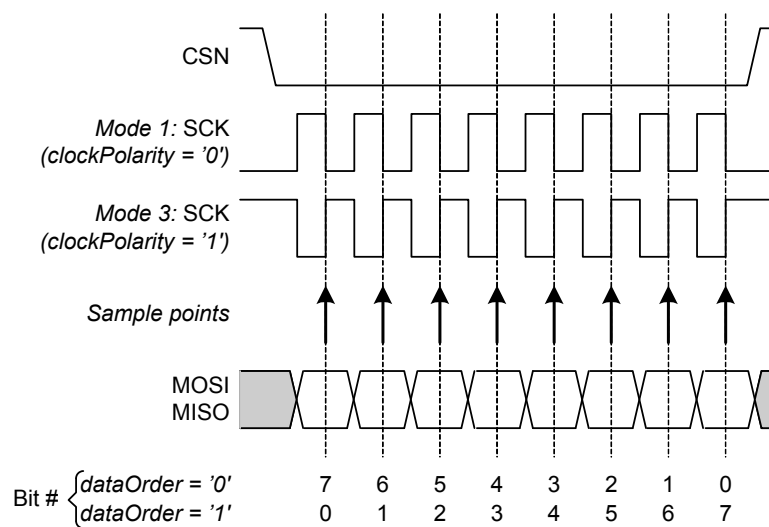
SPI mode	clockPolarity	clockPhase	Clock shift edge		Clock sample edge	
0	0	0	Trailing	Falling	Leading	Rising
1	0	1	Leading	Rising	Trailing	Falling
2	1	0	Trailing	Rising	Leading	Falling
3	1	1	Leading	Falling	Trailing	Rising

**Table 75** SPI modes



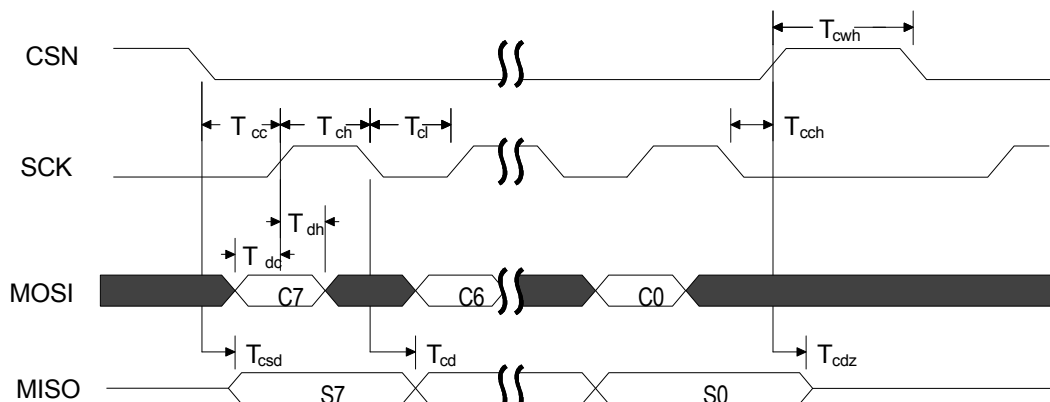


**Figure 54** SPI Modes 0 and 2:  $\text{clockPhase} = '0'$ . One byte transmission.



**Figure 55** SPI Modes 1 and 3:  $\text{clockPhase} = '1'$ . One byte transmission.

SPI timing is given in **Figure 56** and in **Table 76** and **Table 77**



**Figure 56** SPI timing diagram. One byte transmission.

Parameters	Symbol	Min	Max	Units
Data to SCK Setup	Tdc	2		ns
SCK to Data Hold	Tdh	2		ns
CSN to Data Valid	Tcsd		38	ns
SCK to Data Valid	Tcd		55	ns
SCK Low Time	Tcl	40		ns
SCK High Time	Tch	40		ns
SCK Frequency	Fsck	0	8	MHz
SCK Rise and Fall	Tr,Tf		100	ns
CSN to SCK Setup	Tcc	2		ns
SCK to CSN Hold	Tcch	2		ns
CSN Inactive time	Tcwh	50		ns
CSN to Output High Z	Tcdz		38	ns

**Table 76** SPI timing parameters ( $C_{Load} = 5pF$ )

Parameters	Symbol	Min	Max	Units
Data to SCK Setup	Tdc	2		ns
SCK to Data Hold	Tdh	2		ns
CSN to Data Valid	Tcsd		42	ns
SCK to Data Valid	Tcd		58	ns
SCK Low Time	Tcl	40		ns
SCK High Time	Tch	40		ns
SCK Frequency	Fsck	0	8	MHz
SCK Rise and Fall	Tr,Tf		100	ns
CSN to SCK Setup	Tcc	2		ns

Parameters	Symbol	Min	Max	Units
SCK to CSN Hold	Tcch	2		ns
CSN Inactive time	Tcwh	50		ns
CSN to Output High Z	Tcdz		42	ns

**Table 77** SPI timing parameters ( $C_{Load} = 10\text{pF}$ )

## 17 ADC

nRF31512 includes a general purpose ADC with two input channels. The ADC contains an internal 3\*1.2 V reference. It can be operated in a single step mode with sampling under software control.

### 17.1 Features

- 8 or 10 bit resolution
- Two input channels
- Single ended input
- Full-scale range is 3\* internal reference voltage
- Single step mode with conversion time down to 20  $\mu$ s
- Mode for measuring supply voltage

### 17.2 Functional description

The ADC is a first-order single-ended incremental A/D converter. The converter supports 8 and 10 bit conversion. The chosen ADC topology has a very good DC accuracy and uses the calibrated bandgap reference as reference voltage so it is optimum for battery monitoring purposes. It can also be used to measure voltages on port P0.4 to P0.5.

The conversion time is dependant on the resolution and can be expressed as  $t_{CLK} * 2^N + 4 \mu s$  where N is the number of bits in the result. This gives a conversion time of ~20  $\mu$ s for 8 bit resolution and 68  $\mu$ s for 10 bit resolution.

To perform a 10-bit battery voltage measurement, proceed as follows:

```
Write ADCCON3 = 0x80      // 10 bit conversion
Write ADCCON1 = 0x80      // Start conversion, channel 0(P0.4)
Wait for ADCCON1[6] = '0' // Busy flag, goes low when conversion finished
Read ADCDATH and ADCDATL
Recombine 10-bit result C = ADCDATH + 256*ADCDATL
Calculate battery voltage: vbat = C*vstep [V]    // vstep = 3.52mV for 10 bit resolution
// vstep = 14.1mV for 8 bit resolution
```

**Note:** ADC is automatically shut off after conversion so no further action is necessary.

#### 17.2.1 Activation

A write operation to the ADCCON1 register automatically starts a conversion, provided that the pwrup bit is set. If the ADC is busy, any write to pwrup will be ignored. Write operations to ADCCON3 do not start a conversion. It is not advisable to change these registers while the ADC is busy.

#### 17.2.2 Input selection

The ADC supports up to two external and one internal input channels, for single ended measurements. Input channel is selected with the chsel bits. Channel 0 to 1 (AIN4-AIN5) are external inputs applied through port pins. Channel 14 is an internally generated input equal to VDD. See **chapter 15 on page 117** for a description of the mapping between port pins and AIN4-AIN5.

The internally generated VDD input may be used for supply voltage measurement or calibration of offset and gain error.

### 17.2.3 Reference

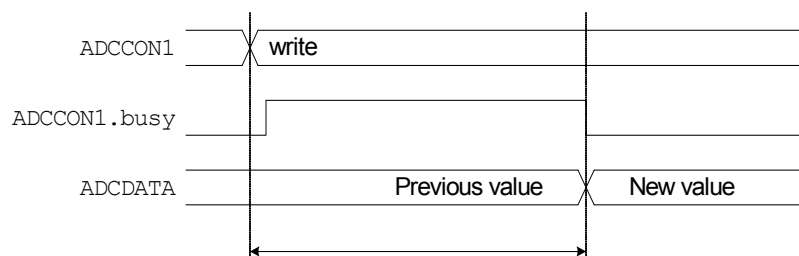
Full-scale range is three times the internal bandgap reference (nominally  $3 \times 1.2 \text{ V} = 3.6 \text{ V}$ ).

### 17.2.4 Resolution

The ADC can do 8 or 10 bit conversions. Configure the `resol` bits to set resolution.

### 17.2.5 Conversion modes

In single step mode the ADC performs one conversion and then stops.



**Figure 57** Timing diagram for single step conversion

**Figure 57** illustrates the timing of a single step conversion. The conversion is started by writing to the `ADCCON1` register. The busy bit is set to '1' immediately afterwards and cleared again when the conversion result becomes available in the `ADCDATH/ADCDATL` registers. An interrupt to the MCU (`ADCIRQ`) is also generated at the end of conversion.

### 17.2.6 Output data coding

The ADC uses straight binary coding for single ended conversions. An input voltage  $\leq 0\text{V}$  is represented by all zeroes (000...00), and an input voltage  $\geq V_{\text{REF}}$  by all ones (111...11). Midscale is represented by a one followed by all zeroes (100...00).

## 17.2.7 SFR registers

The ADC is interfaced to the MCU through five registers; ADCCON1, ADCCON3, ADCDATH and ADCDATL. ADCCON1 and ADCCON3 contain configuration settings and status bits. The conversion result is contained in the ADCDATH and ADCDATL registers.

Addr	Bit	Name	RW	Function	Reset value: 0x00
0xD3	7	pwrup	RW	Power-up control: 0: Power down ADC 1: Power up ADC and configure selected pin(s) as analog input. A write will start a conversion and will always read back as 0.	
	6	busy	R	ADC busy flag: 0: No conversion in progress 1: Conversion in progress The <b>busy</b> bit is cleared when a conversion result becomes available in the ADCDATH / ADCDATL registers.	
	5:2	chsel	RW	Input channel select: 0100: AIN4 0101: AIN5 : 1110: VDD pad Other values reserved, do not use.	
	1:0	refsel	RW	Reference select: 00: 3* (Internal 1.2V reference)=3.6V Other values reserved, do not use.	

**Table 78** ADCCON1 register

Addr	Bit	Name	RW	Function	Reset value: 0x00
0xD1	7:6	resol	RW	ADC resolution: 00: Reserved, do not use 01: 8 bits 10: 10 bits 11: Reserved, do not use	
	5:0	-	-	Reserved, MBZ	

*Table 79 ADCCON3 register*

Addr	Bit	Name	RW	Function	Reset value: 0x00
0xD4	7:0	-	R	Most significant byte of result ADCDATA	

*Table 80 ADCDATH register*

Addr	Bit	Name	RW	Function	Reset value: 0x00
0xD5	7:0	-	R	Least significant byte of result ADCDATA	

*Table 81 ADCDATL register*

## 18 Absolute maximum ratings

Maximum ratings are the extreme limits to which the device can be exposed without permanently damaging it. Exposure to absolute maximum ratings for prolonged periods of time may affect device reliability.

The device is not guaranteed to operate properly at the maximum ratings.

Operating conditions	Minimum	Maximum	Units
<b>Supply voltages</b>			
VDD	-0.3	+3.6	V
VSS		0	V
<b>I/O pin voltages</b>			
PROG/VPP	-0.3	+6.5	V
Other I/O	-0.3	VDD +0.3, max 3.6	V
<b>Temperatures</b>			
Storage temperature	- 40	+125	°C

**Table 82** Absolute maximum ratings

**Note:** Stress exceeding one or more of the limiting values may cause permanent damage to the device.

### Attention!

Observe precaution for handling Electrostatic Sensitive Device.

HBM (Human Body Model): Class 1C

CDM (Charged Device Model): Class II





## 19 Operating condition

Symbol	Parameter	Notes	Min.	Typ.	Max.	Units
VDD	Supply voltage		1.9	3.0	3.3	V
t <sub>R_VDD</sub>	Supply rise time (0V to 1.9V)	1	1 μs		50 ms	μs and ms
T <sub>A</sub>	Operating temperature		-10		+60	°C

1. The on-chip power-on reset circuitry may not function properly for rise times outside the specified interval.

**Table 83** *Operating conditions*

## 20 Electrical specifications

This section contains electrical and timing specifications.

Conditions: VDD = 3.0V, T<sub>A</sub> = –10°C to +60°C (unless otherwise noted)

Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
V <sub>IH</sub>	Input high voltage		0.7xVDD		VDD	V
V <sub>IL</sub>	Input low voltage		VSS		0.3x VDD	V
V <sub>OH</sub>	Output high voltage (std. drive, 0.5mA)		VDD - 0.3		VDD	V
V <sub>OH</sub>	Output high voltage (high-drive, 5mA)	1	VDD - 0.3		VDD	V
V <sub>OL</sub>	Output low voltage (std. drive, 0.5mA)		VSS		0.3	V
V <sub>OL</sub>	Output low voltage (high-drive, 5mA)	1	VSS		0.3	V
R <sub>PU</sub>	Pull-up resistance		11	13	16	kΩ
R <sub>PD</sub>	Pull-down resistance		11	13	16	kΩ

1. Maximum number of pins with 5mA high drive is 3.

**Table 84** Digital inputs/outputs

Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
<b>General RF conditions</b>						
f <sub>OP</sub>	Operating frequency	1	2400		2525	MHz
PLL <sub>res</sub>	PLL Programming resolution			1		MHz
f <sub>XTAL</sub>	Crystal frequency			16		MHz
Δf <sub>250</sub>	Frequency deviation @ 250 kbps			±170		kHz
Δf <sub>1M</sub>	Frequency deviation @ 1 Mbps			±170		kHz
Δf <sub>2M</sub>	Frequency deviation @ 2 Mbps			±320		kHz
R <sub>GFSK</sub>	Air data rate	2	250		2000	kbps
F <sub>CHANNEL 1M</sub>	Non-overlapping channel spacing @ 250 kbps/1 Mbps)	3		1		MHz
F <sub>CHANNEL 2M</sub>	Non-overlapping channel spacing @ 2 Mbps			2		MHz
<b>Transmitter operation</b>						
P <sub>RF</sub>	Maximum output power	4		0	+4	dBm
P <sub>RFC</sub>	RF power control range		16	18	20	dB
P <sub>RFCR</sub>	RF power accuracy				±4	dB
P <sub>BW2</sub>	20dB bandwidth for modulated carrier (2 Mbps)			1800	2000	kHz
P <sub>BW1</sub>	20dB bandwidth for modulated carrier (1 Mbps)			950	1100	kHz
P <sub>BW250</sub>	20dB bandwidth for modulated carrier (250 kbps)			700	800	kHz

Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
$P_{RF1.2}$	1 <sup>st</sup> Adjacent Channel Transmit Power 2 MHz (2Mbps)				-20	dBc
$P_{RF2.2}$	2 <sup>nd</sup> Adjacent Channel Transmit Power 4 MHz (2 Mbps)				-45	dBc
$P_{RF1.1}$	1 <sup>st</sup> Adjacent Channel Transmit Power 1 MHz (1 Mbps)				-20	dBc
$P_{RF2.1}$	2 <sup>nd</sup> Adjacent Channel Transmit Power 2 MHz (1 Mbps)				-40	dBc
$P_{RF1.250}$	1 <sup>st</sup> Adjacent Channel Transmit Power 1 MHz (250 kbps)				-25	dBc
$P_{RF2.250}$	2 <sup>nd</sup> Adjacent Channel Transmit Power 2 MHz (250 kbps)				-40	dBc
<b>Receiver operation</b>						
$RX_{MAX}$	Maximum received signal at < 0.1% BER			0		dBm
$RX_{SENS}$	Sensitivity (0.1% BER) @ 2 Mbps			-82		dBm
$RX_{SENS}$	Sensitivity (0.1% BER) @ 1 Mbps			-85		dBm
$RX_{SENS}$	Sensitivity (0.1% BER) @ 250 kbps	5		-94		dBm
<b>RX selectivity with nRF24L01 equal modulation on interfering signal (Pin = -67dBm for wanted signal)</b>						
$C/I_{CO}$	C/I co-channel (2 Mbps) (modulated carrier)			11		dBc
$C/I_{1ST}$	1 <sup>st</sup> ACS (Adjacent Channel Selectivity), C/I 2MHz (2 Mbps)			4		dBc
$C/I_{2ND}$	2 <sup>nd</sup> ACS, C/I 4 MHz (2 Mbps)			-18		dBc
$C/I_{3RD}$	3 <sup>rd</sup> ACS, C/I 6 MHz (2 Mbps)			-21		dBc
$C/I_{Nth}$	N <sup>th</sup> ACS, C/I $f_i > 12$ MHz (2 Mbps)			-40		dBc
$C/I_{Nth}$	N <sup>th</sup> ACS, C/I $f_i > 36$ MHz (2 Mbps)			-48		dBc
$C/I_{CO}$	C/I co-channel (1 Mbps)			14		dBc
$C/I_{1ST}$	1 <sup>st</sup> ACS, C/I 1 MHz (1 Mbps)			11		dBc
$C/I_{2ND}$	2 <sup>nd</sup> ACS, C/I 2 MHz (1 Mbps)			-21		dBc
$C/I_{3RD}$	3 <sup>rd</sup> ACS, C/I 3 MHz (1 Mbps)			-30		dBc
$C/I_{Nth}$	N <sup>th</sup> ACS, C/I $f_i > 6$ MHz (1 Mbps)			-40		dBc
$C/I_{Nth}$	N <sup>th</sup> ACS, C/I $f_i > 25$ MHz (1 Mbps)			-50		dBc
$C/I_{CO}$	C/I co-channel (250 kbps)			7		dBc
$C/I_{1ST}$	1 <sup>st</sup> ACS, C/I 1 MHz (250 kbps)			-7		dBc
$C/I_{2ND}$	2 <sup>nd</sup> ACS, C/I 2 MHz (250 kbps)			-34		dBc
$C/I_{3RD}$	3 <sup>rd</sup> ACS, C/I 3 MHz (250 kbps)			-39		dBc
$C/I_{Nth}$	N <sup>th</sup> ACS, C/I $f_i > 6$ MHz (250 kbps)			-50		dBc
$C/I_{Nth}$	N <sup>th</sup> ACS, C/I $f_i > 25$ MHz (250 kbps)			-60		dBc

Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
<b>RX intermodulation performance in line with Bluetooth specification version 2.0, 4<sup>th</sup> November 2004, page 42</b>						
P_IM(6) @ 2Mbps	Input power of IM interferers at 6 and 12 MHz distance from wanted signal	6		-42		dBm
P_IM(8) @ 2Mbps	Input power of IM interferers at 8 and 16 MHz distance from wanted signal	7		-40		dBm
P_IM(10) @ 2Mbps	Input power of IM interferers at 10 and 20 MHz distance from wanted signal	7		-39		dBm
P_IM(3) @ 1Mbps	Input power of IM interferers at 3 and 6 MHz distance from wanted signal	7		-39		dBm
P_IM(4) @ 1Mbps	Input power of IM interferers at 4 and 8 MHz distance from wanted signal	7		-39		dBm
P_IM(5) @ 1Mbps	Input power of IM interferers at 5 and 10 MHz distance from wanted signal	7		-39		dBm
P_IM(3) @ 250kbps	Input power of IM interferers at 3 and 6 MHz distance from wanted signal	7		-36		dBm
P_IM(4) @ 250kbps	Input power of IM interferers at 4 and 8 MHz distance from wanted signal	7		-36		dBm
P_IM(5) @ 250kbps	Input power of IM interferers at 5 and 10 MHz distance from wanted signal	7		-36		dBm
<b>ADC</b>						
V <sub>OS</sub>	Offset error	7 8		+/- 3		% FS
ε <sub>G</sub>	Gain error	9		+/- 3		% FS
V <sub>REF_INT</sub>	Internal reference voltage			1.2		V
TC <sub>REF_INT</sub>	Internal reference voltage drift			300		ppm/°C
<b>16 MHz crystal</b>						
f <sub>NOM</sub>	Nominal frequency (parallel resonant)			16.000		MHz
f <sub>TOL</sub>	Frequency tolerance	10 11			±60	ppm
C <sub>L</sub>	Load capacitance			9	16	pF
C <sub>0</sub>	Shunt capacitance				7	pF
ESR	Equivalent series resistance			50	100	Ω
P <sub>D</sub>	Drive level				100	μW
L <sub>S</sub>	Equivalent series inductance	12		30		mH
<b>16 MHz RC oscillator</b>						
f <sub>NOM</sub>	Nominal frequency			16		MHz
f <sub>TOL</sub>	Frequency tolerance			±1	±5	%
<b>32 kHz RC oscillator</b>						
f <sub>NOM</sub>	Nominal frequency			32.8		kHz
f <sub>TOL</sub>	Frequency tolerance			±1	±10	%

Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
<b>Power-Fail Comparator</b>						
V <sub>POF</sub>	Nominal thresholds (falling supply voltage)		2.0, 2.1, 2.2, 2.3			V
V <sub>TOL</sub>	Threshold voltage tolerance				±5	%
V <sub>HYST</sub>	Threshold voltage hysteresis			50		mV

- Usable band is determined by local regulations.
- Data rate in each burst on-air.
- The minimum channel spacing is 1 MHz.
- Antenna load impedance =  $15\ \Omega + j88\ \Omega$ .
- For 250 kbps sensitivity, frequencies which are integer multiples of 16 MHz (2400, 2416 and so on) sensitivity are reduced.
- Wanted signal level at Pin = -64 dBm. Two interferers with equal input power are used. The interferer closest in frequency is unmodulated, the other interferer is modulated equal with the wanted signal. The input power of interferers where the sensitivity equals BER = 0.1% is presented.
- Measured with 10-bit resolution, single-ended input and 3\*VREF\_INT as reference
- Defined as the deviation of the first code transition (000...000) to (000...001) from the ideal.
- Defined as the deviation of the last code transition (111...110) to (111...111) from the ideal, after correcting for offset error.
- Includes initial accuracy, stability over temperature, aging and frequency pulling due to incorrect load capacitance.
- Frequency regulations in certain regions set tighter requirements on frequency tolerance (for example Japan and South Korea max ±50 ppm).
- Startup time from power down to standby mode depends on the L<sub>s</sub> parameter.

*Table 85 Electrical specifications*

## 20.1 OTP memory

Characteristic	Symbol	Conditions	Min.	Typ.	Max	Unit
Programming voltage	VPP	-10 to +60 °C	6.25	6.35	6.45	V
Data retention	Tret	60 °C	>10			years

*Table 86 OTP memory characteristics*

Name	Size	Unit
OTP memory MainBlock	17397	bytes
OTP InfoPage	10	bytes

*Table 87 OTP memory size*

## 20.2 Power consumption

The power consumption is always a sum of the current draw from all modules active at the time of measurement and is very application dependent.

To calculate a peak current draw summarize the currents from all modules that can be active at the same time in a given application.

Conditions: VDD = 3.0 V, TA = +25° C

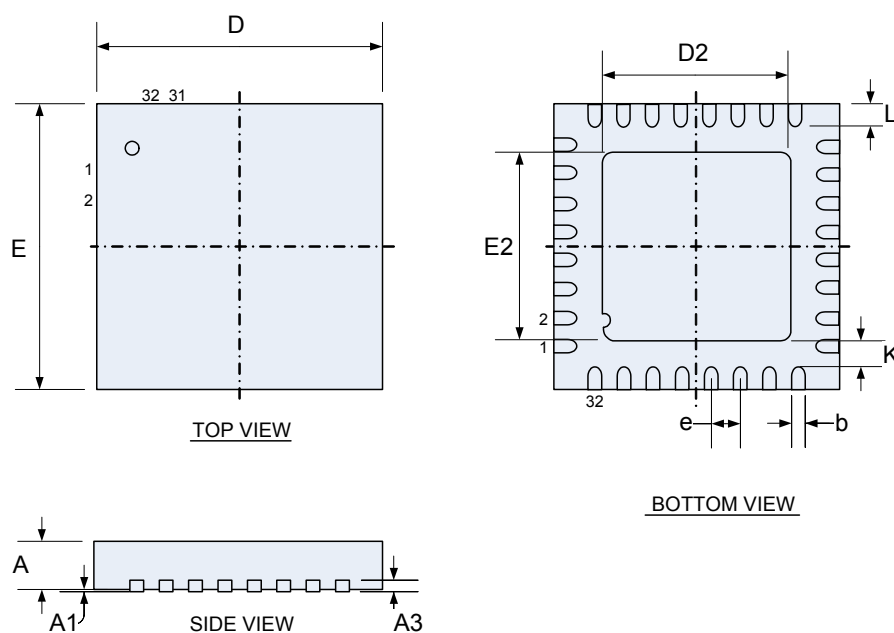
Symbol	Parameter (condition)	Notes	nRF31512 Typ.	Units
<b>Core functions<sup>1</sup></b>				
	Register retention mode, timers off		1	μA
	Register retention mode, timers on (CLKLF from RCOSC32K)		2	μA
	Register retention mode, timers on (CLKLF from RCOSC32K, XOSC16M running)		50	μA
	Register retention mode, timers on (CLKLF synthesized from XOSC16M)		150	μA
	Standby mode (XOSC16M running)		1	mA
	Active mode (8 MHz MCU clock, 4 MIPS)		4	mA
<b>Peripherals</b>				
	OTP byte write		3.8	mA
	RF transceiver in TX mode (P <sub>OUT</sub> = 0 dBm)	2	11.1	mA
	RF transceiver in TX mode (P <sub>OUT</sub> = -6 dBm)		8.8	mA
	RF transceiver in TX mode (P <sub>OUT</sub> = -12 dBm)		7.3	mA
	RF transceiver in TX mode (P <sub>OUT</sub> = -18 dBm)		6.8	mA
	RF transceiver in TX mode (P <sub>OUT</sub> = -6 dBm) Average current with ShockBurst™	3	0.12	mA
	RF transceiver during TX settling	4	7.8	mA
	RF transceiver in RX mode (2 Mbps)		13.3	mA
	RF transceiver in RX mode (1 Mbps)		12.9	mA
	RF transceiver in RX mode (250 kbps)		12.4	mA
	RF transceiver during RX settling	5	8.7	mA
	ADC when busy		0.2	mA

1. Please note that all pins must be set to inputs, and the pinMode input mode bits (refer to **Table 71 on page 123**, **Table 72 on page 124**, **Table 76 on page 131**, and **on page 125**) must be set to 11 (Digital input buffer off) if the pins are not controlled externally.
2. Peak current at 1 Mbps. Antenna load impedance = 15 Ω + j88Ω.
3. Average data rate 10 kbps and full packets.
4. Average current consumption for TX startup (130 μs), and when changing mode from RX to TX (130 μs).
5. Average current consumption for RX startup (130 μs), and when changing mode from TX to RX (130 μs).

**Table 88** Power consumption

## 21 Mechanical specifications

The data for the applicable QFN packets is shown below. For information regarding ordering codes and package markings, see *Table 92 on page 147*.



**Figure 58** QFN32 pin 5x5 mm

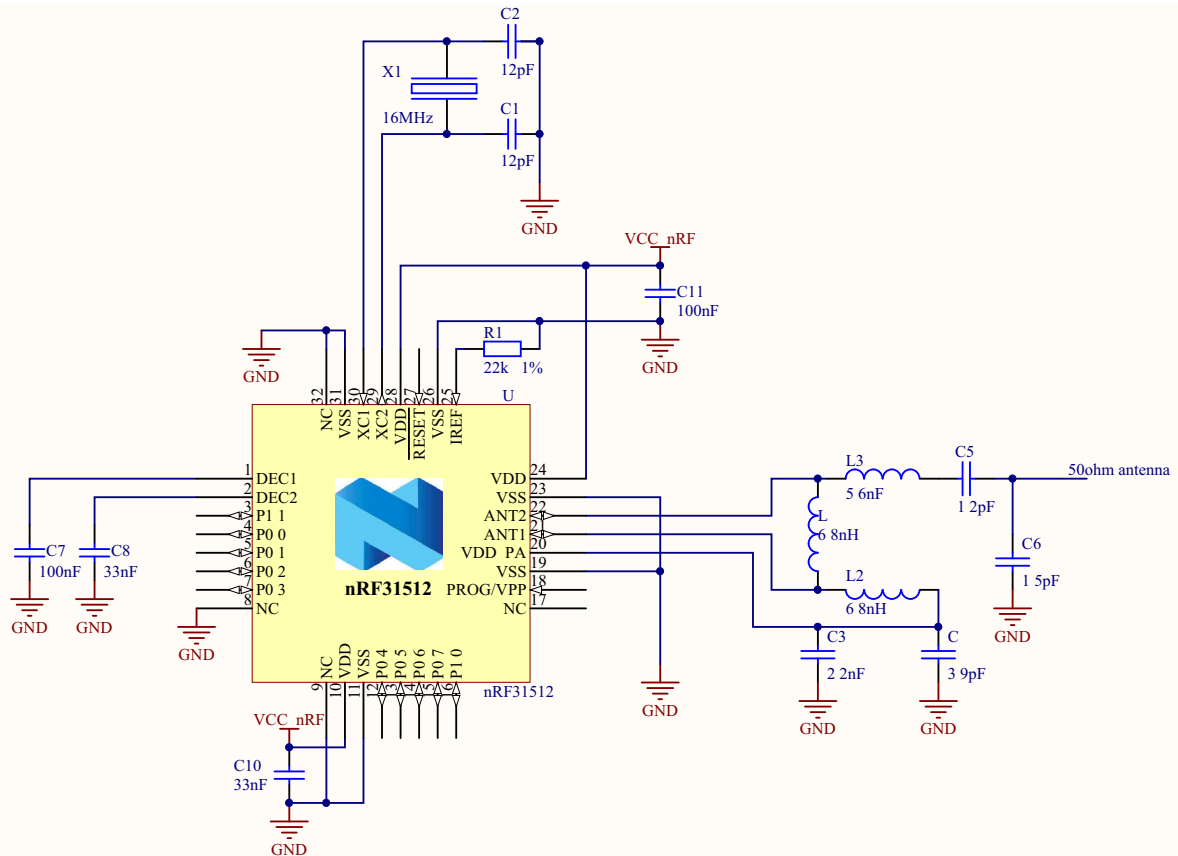
Package	A	A1	A3	b	D, E	D2, E2	e	K	L	
QFN32	0.80	0.00		0.18		3.50		0.20	0.35	Min
	0.85	0.02	0.20	0.25	5.0	3.60	0.5		0.40	Typ
	0.90	0.05		0.30		3.70			0.45	Max

**Table 89** QFN32 dimensions in mm

## 22 Reference circuits

### 22.1 nRF31512, 5x5 mm QFN32

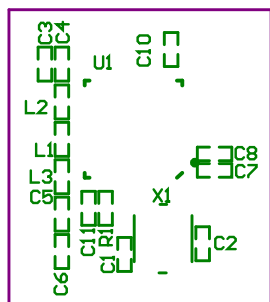
#### 22.1.1 Schematic



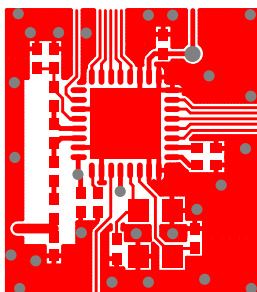
**Figure 59** Schematic for nRF31512



## 22.1.2 Layout

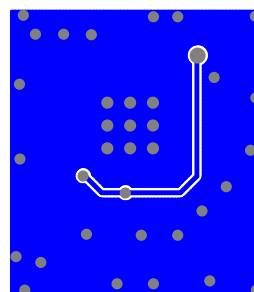


Top silk screen



Top view

No components  
in bottom layer



Bottom view

## 22.1.3 Bill Of Materials

Designator	Value	Footprint	Comment
C1, C2	12 pF	0402	NPO
C3	2.2 nF	0402	X7R
C4	3.9 pF	0402	NPO
C5	1.2 pF	0402	NP0 +/- 0.1 pF
C6	1.5 pF	0402	NP0 +/- 0.1 pF
C7, C11	100 nF	0402	X7R
C8, C10	33 nF	0402	X7R
L1, L2	6.8 nH	0402	Chip inductor +/-5%
L3	5.6 nF	0402	Chip inductor +/-5%
R1	22 k	0402	1%
U1	nRF31512	QFN32	QFN32 5x5 mm Package
X1	16 MHz	3.2 mm x 2.5 mm	TSX3225, 16 MHz, CI=9 pF, +/- 10 ppm

Table 90 Bill of Materials

## 23 Ordering information

### 23.1 Package marking

N	R	F		A	X
P	P	P	P	P	Z
Y	Y	W	W	L	L

#### 23.1.1 Abbreviations

Abbreviation	Definition
PPPPP	Product number, 31512
X	"X" grade, that is, Engineering Samples (optional)
Z	QFN package type "C" = 32 pin
YY	Two digit Year number
WW	Two digit week number
LL	Two letter wafer lot number code
A	Build Code, that is, unique code for production sites, package type and test platform

**Table 91** Abbreviations

## 23.2 Ordering codes

Ordering code	Package	Container	MOQ	MSL level
nRF31512-O17Q32-T	5x5 mm 32-pin QFN, lead free (green)	Tray	490	2
nRF31512-O17Q32-R7	5x5 mm 32-pin QFN, lead free (green)	Tape-and-reel 7"	1500	2
nRF31512-O17Q32-R	5x5 mm 32-pin QFN, lead free (green)	Tape-and-reel 13"	4000	2
nRF31512-O17Q32-S	5x5 mm 32-pin QFN, lead free (green)	Sample box	5	2

**Table 92** Ordering codes

The devices will not have separate development kits. For development use nRF24LE1 devices with flash memory. Socket programming adapters for the nRF31512 are available from Nordic Semiconductor. These are intended for use in engineering samples and pilot runs. For volume production use commercially available OTP programming tools.

Type Number	Description
nRF6700	nRFgo Starter Kit
nRF24LE1-F16Q32-DK	nRFgo Development Kit for nRF24LE1 5x5 mm 32 pin QFN (requires nRFgo Starter Kit)
nRF6705	nRFgo nRF31512 32-pin Programming Adapter Kit (requires nRFgo Starter Kit)

**Table 93** Development tools

## 24 Glossary

Term	Description
ACK	Acknowledgement
ADC	Analog to digital converter
ART	Auto Re-Transmit
BOR	Brown-Out Reset
CE	Chip Enable
CLK	Clock
CRC	Cyclic Redundancy Check
CSN	Chip Select NOT
ESB	Enhanced ShockBurst™
GFSK	Gaussian Frequency Shift Keying
IRQ	Interrupt Request
ISM	Industrial-Scientific-Medical
LNA	Low Noise Amplifier
LSB	Least Significant Bit
LSByte	Least Significant Byte
Mbps	Megabit per second
MCU	Microcontroller
MISO	Master In Slave Out
MOQ	Minimum Order Quantity
MOSI	Master Out Slave In
MSB	Most Significant Bit
MSByte	Most Significant Byte
MSL	Moisture Sensitivity Level according to JEDEC classification
NV	Non-Volatile (memory)
PCB	Printed Circuit Board
PER	Packet Error Rate
PID	Packet Identity Bits
PLD	Payload
POF	Power Fail
POR	Power On Reset
PRX	Primary RX
PTX	Primary TX
PWR_DWN	Power Down
PWR_UP	Power Up
QFN	Quad Flat No lead Package
RCOSC16M	16 MHz RC oscillator
RCOSC32K	32 KHz RC oscillator
RX	Receive
RX_DR	Receive Data Ready

---

SPI	Serial Peripheral Interface
TX	Transmit
TX_DS	Transmit Data Sent
XOSC16M	16 MHz crystal oscillator

*Table 94 Glossary*